ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institute of Theoretical Computer Science
Bernd Gärtner, Rasmus Kyng, Angelika Steger, David Steurer, Emo Welzl

Algorithms, Probability, and Computing — Special Assignment 2 — HS23

- Write your solutions using a computer, where we strongly recommend to use LaTeX. **We do not grade hand-written solutions.**
- The solution is due on **December 5th, 2023** by **2 pm**. Please submit one file per exercise on Moodle.
- For geometric drawings that can easily be integrated into LaTeX documents, we recommend the drawing editor IPE, retrievable at `http://ipe.otfried.org` or through various package managers.
- Write short, simple, and precise sentences.
- This is a theory course, which means: if an exercise does not explicitly say "you do not need to prove your answer" or "justify intuitively", then a formal proof is **always** required. You can of course refer in your solutions to the lecture notes and to the exercises, if a result you need has already been proved there.
- We would like to stress that the ETH Disciplinary Code applies to this special assignment as it constitutes part of your final grade. The only exception we make to the Code is that we encourage you to verbally discuss the tasks with your colleagues. However, you need to include a list of all of your collaborators in each of your submissions. It is strictly prohibited to share any (hand)written or electronic (partial) solutions with any of your colleagues. We are obligated to inform the Rector of any violations of the Code.
- There will be two special assignments this semester. Both of them will be graded and the average grade will contribute 20% to your final grade.
- As with all exercises, the material of the special assignments is relevant for the (midterm and final) exams.

Exercise 1 10 points

(*Flow in bipartite graph*)
Let $G = (V, E)$ be bipartite graph where $V = S \cup T$, $|S| = n$, and $|T| = m$. Furthermore let $s : S \mapsto \mathbb{R}_{\geq 0}$, $d : T \mapsto \mathbb{R}_{\geq 0}$ be positive functions.

All the edges $e = (u, v)$ in G have one endpoint in S and one endpoint in T and G represents a network that connect nodes $i \in S$ with the nodes $j \in T$. Suppose that each node $i \in S$ comes with a quantity $s(i)$ of available resources and each node $j \in T$ comes with a demand of resources $d(j)$. We want to figure out whether there is a way to route the resources from the nodes in S to the nodes in T that the satisfies the demand. For simplicity, we make the assumption that $\sum_i s(i) = \sum_j d(j)$, i.e. the total quantity of resources available matches the total demand.

Create variable x_{ij} for each edge $(i, j) \in E$ and consider the following linear program.

$$\max \sum_{e \in E} 0 \ x_e$$

such that:

$$\sum_j x_{ij} \leq s(i), \quad \forall i \in S;$$

$$\sum_i x_{ij} \geq d(j), \quad \forall j \in T;$$

$$x_{i,j} \geq 0$$

(a) Prove that there is a way to match the resources to the demands if and only if the linear program is feasible.

(b) Write the dual of the linear program above.

(c) Prove that the first linear program has a solution if and only if the following condition holds. For each $Q \subseteq S$,

$$\sum_{i \in Q} s(i) \leq \sum_{j : \exists (i,j) \in E, i \in Q} d(j)$$

and for every $Q \subseteq T$,

$$\sum_{j \in Q} d(j) \leq \sum_{i : \exists (i,j) \in E, j \in Q} s(i)$$

.

Hint: look at Theorem 4.6 from the lecture notes.

Answer to Exercise 1

(a) Suppose that the LP is feasible, then there exists a solution $\{x_{ij}\}$ that satisfy all the constraints. We send a quantity x_e of resources on each edges e of the graph. The first constraint guarantees that for each node $i \in S$ the amount of resources leaving that node is at most $s(i)$. The second constraint guarantees that each node $j \in T$ obtain at least $d(j)$ resources.
Conversely, suppose that there exists a way to route the resources in the network that satisfies the requirements. Assign to the variable x_{ij} the amount of resources that are sent from node $i \in S$ to node $j \in T$. Then, $\{x_{ij}\}$ is a feasible solution to the LP, hence the LP is feasible.

(b) Let p_i, q_j for $i \in S$ and $j \in T$ be the dual variables. The dual program is:

$$\min \sum_{i \in S} p_i s(i) - \sum_{j \in T} q_j d(j)$$

such that:

$$p_i - q_j \geq 0, \quad \forall (i,j) \in E.$$
$$p_i \geq 0$$
$$q_j \geq 0$$

(c) Let $K = \sum_i s(i) = \sum_j d(j)$. First, suppose that one of the condition are not met, for example there exists $\bar{i} \in S$ such that $\sum_{j:(\bar{i},j) \in E)} d(j) < s(\bar{i})$. Fix $M \in \mathbb{R}_{\geq 0}$ and consider the assignment

$$p_i = \begin{cases} 0 & i = \bar{i}, \\ M, & i \neq \bar{i}; \end{cases}$$

$$q_j = \begin{cases} 0, & (\bar{i}, j) \in E, \\ M, & (\bar{i}, j) \notin E. \end{cases}$$

This is a feasible solution of the dual program and the objective function is

$$\sum_{i \in S} p_i d(i) - \sum_{j \in T} q_j s(j) = (K - s(\bar{i}))M - (K - \sum_{j:(\bar{i},j) \in E)} d(j))M = (\sum_{j:(\bar{i},j) \in E} d(j) - s(\bar{i}))M$$

If we now let $M \rightarrow +\inf$ we see that the dual program is unbounded.
If instead, we have that there exists $\bar{j}$ such that $\sum_{i:(i,\bar{j}) \in E} s(i) \geq d(\bar{j})$, an analogous conclusion can be obtained by looking at the assignment

$$p_i = \begin{cases} M, & (i, \bar{j}) \in E), \\ 0, & (i, \bar{j}) \notin E; \end{cases}$$

$$q_j = \begin{cases} M, & j = \bar{j}, \\ 0, & j \neq \bar{j}. \end{cases}$$

Finally, Theorem 4.6 from the lecture notes states that since the dual program is unbounded, the primal program cannot be feasible.

On the other hand, suppose that the condition are satisfied. Then, we claim that the assignment $p_i = 0$, for all $i \in S$ and $q_j = 0$ for all $j \in T$ is the optimal solution of the dual program . We apply again Theorem 4.6 from the lecture notes that states that since the dual program is feasible and bounded, the primal program has a (optimal) solution with cost 0 (not surprisingly). We now prove that the claimed assignment is indeed optimal. To do so, assume that you are being given an assignment that is optimal but has $q_j \neq 0$ for some j. Otherwise, we can clearly set all $p_i = 0$ to obtain an optimal solution. But then let $Q \subset T$ be the set of $j \in T$ with $q_j > 0$. Let ϵ be the smallest value among them, and let $N(Q)$ denote their neighbors in S. Then, we remove ϵ from all q_j for $j \in Q$ and all p_i for $i \in N(Q)$. Clearly, this is still a feasible solution since every non-zero q_j got reduced by ϵ. But by the second condition, the objective value of this solution is at least as good as the original one. Now, we iterate this argument until all q_j are 0.

Exercise 2 25 points

(*Grading scheduling*)

The APC exam consists of n exercises, and the course has m teaching assistants that partake in grading. Based on their skills, each TA i takes t_{ij} hours to grade exercise number j. We aim to grade all the exercises in the least time possible such that feedback can be provided in a timely manner[1]. In other words, let x_{ij} an indicator variable that attains value 1 if exercise i get assigned to TA j and zero otherwise. Then we want to minimize $\max_j \sum_i x_{ij} t_{ij}$. Consider the following integer program[2].

$$\min T$$

$$\sum_i x_{ij} \geq 1, \quad \forall j$$

$$T \geq \sum_j x_{ij} t_{ij}, \quad \forall i$$

$$x_{ij} \in \{0, 1\}, \quad \forall i, j$$

(a) Argue that the integer program finds the optimal assignment of exercises to TAs.

(b) Relax the program above to an LP (substitute $x_{ij} \in \{0,1\}$ with $x_{ij} \in [0,1]$) and prove that the integrality gap of the program above is at least m. In other words, find an example of $t_{i,j}$ so that the ratio between the best solution of the integer program above and the relaxed program is at least m.

We now modify the algorithm to get a much better integral solution via an intricate rounding scheme. Let $\tilde{t}$ be a fixed value and consider the relaxation of the LP as in (b) above with the extra constraint that

$$x_{ij} = 0, \quad \text{if } t_{ij} > \tilde{t}$$

for some threshold value $\tilde{t}$. When $\tilde{t}$ is chosen appropriately, this captures the intuition that questions should not be allocated to TAs that know very little about the topic.

(c) Show that the LP is feasible for some value $\tilde{t} = t_{ij}$.

Now, suppose that this new problem is feasible for some $\tilde{t}$ and and let x be an optimal but fractional solution x with cost $T(\tilde{t})$. We want to show that it is possible to round x into an integral solution of cost at most $T(\tilde{t}) + \tilde{t}$.

We let $q_i = \lceil \sum_j x_{ij} \rceil$. This intuitively corresponds to some educated guess of the number questions TA i is supposed to grade. To allocate questions to TAs, we build a bipartite graph with TAs on the left side, and questions on the right side. TA i appears q_i times on the left side, which encodes that they can be given up to q_i questions to grade. We refer to the copies of TA i as $i_1, \dots i_{q_i}$. Each question just appears once and we will refer to its vertex as j.

[1]For simplicity, we assume that each exercise is only graded by one TA.

[2]A integer program is a linear program with the additional constraint that the solution has to be integral. This turns out to be a very powerful modification, that allows encoding NP hard problems, too.

Next, we define a tricky set of indices. We let $\pi_i(j)$ denote the index of the question TA i is j-th slowest at grading. To illustrate this definition, consider the following example with 3 questions where TA i takes two hours to grade question 1 ($p_{i1} = 2$), three hours for question 2 ($p_{i2} = 3$) and just one hour for question 3 ($p_{i3} = 1$). Then $\pi_i(1) = 2$, $\pi_i(2) = 1$ and $\pi_i(3) = 3$.

We then define a bipartite graph by adding the following edges for each TA i.

1. Initialize $c \leftarrow 1$, $j \leftarrow 1$, $r \leftarrow 1$ and $w \leftarrow x_{i\pi_i(1)}$.
2. While ($j \leq n$):
   - If $w \leq r$, add an edge of weight $x_{i\pi_i(j)}$ from exercise $\pi_i(j)$ to TA i_c if $x_{i\pi_i(j)} \neq 0$. Update $r \leftarrow r - w$, $j \leftarrow j + 1$ and $w \leftarrow x_{i\pi_i(j)}$ (with the newly updated j).
   - Else, add an edge of weight $x_{i\pi_i(j)}$ from exercise $\pi_i(j)$ to TA i_c if $x_{i\pi_i(j)} \neq 0$. Update $w \leftarrow w - r$, $r \leftarrow 1$ and $c \leftarrow c + 1$.

Finally, we construct an unweighted bipartite graph by taking all edges of the weighted bipartite graph we built.

(d) Prove that c is bounded by q_i when running the above algorithm for TA i, i.e. we don't attempt to add an edge to a copy of a TA that doesn't exist.

(e) Show that the unweighted bipartite graph built contains a matching with n edges.

Take a matching with n edges in the unweighted bipartite graph we built.

(f) Prove that the matching returned corresponds to an integral solution with value $T(\tilde{t}) + \tilde{t}$ where $T(\tilde{t})$ is the fractional solution obtained with threshold $\tilde{t}$.

(g) Let OPT be the cost of the optimal solution of the integer program, i.e. the time required for the grading with the optimal assignment. Describe a polynomial time algorithm that finds an assignment of the exercise to TAs such that the exam can be graded in time at most 2OPT.
Hint: How do you choose $\tilde{t}$?

Some useful facts.

Definition 1. *Given a graph* $G = (V, E)$, *a* fractional matching *of* G *is a function* $z : E \mapsto [0, 1]$ *such that*

$$\sum_{e \ni i} z(e) \leq 1$$

for all $i \in V$. *The size of the fractional matching is* $\sum_{e \in E} z(e)$.

Fact 2. *Every bipartite graph is a stable graph, i.e. the size of the biggest fractional matching in a bipartite graph is an integer and equals the size of the biggest matching in the graph.*

Answer to Exercise 2

(a) Because of the integrality constraint, the variables x_{ij} are indeed indicator variables. Then, we notice that all the possible assignments of the exercises to the TAs, are a feasible solution to the problem, i.e. the indicator variables satisfy all the constraints. On the other hand, the first set of constraints ensure that each exercise gets assigned to some TA so, each solution of the integral program constitutes a correct assignment of the exercises. Finally, minimizing T minimizes this maximum timespan for each TA, this proves that the optimal solution of the integral program is also an optimal solution to our problem.

(b) Assume that there is just one exercise, but there are m teaching assistants that all take one hour to grade it. Then, the best possible integral solution achieves time $T = 1$, whereas the best fractional solution achieves time $1/m$.

(c) Fix an optimal integral assignment of cost OPT. We prove a stronger version, namely, $\tilde{t}$ is a value of t_{ij} and $\tilde{t}$ can be set as small as the largest $t_{i,j}$ whose $x^{\star}_{i,j} \neq 0$. Notice that we can use a the optimal assignment in order to construct a feasible solution of the relaxed LP of cost OPT.

(d) Follows from the definition of q_i and the algorithm for constructing the bipartite graph. Note that even though we might connect an exercise node with 2 TA copies, we use the variable r to keep track of the remaining capacity of the current TA copy until it reaches saturation, and the variable w to store the fraction of the current exercise we have not yet allocated.

(e) WLOG assume that x is an optimal solution such that all constraints of the type $\sum_i x_{ij} \geq 1$ hold with equality (note that there always exists one). Basically, while constructing the bipartite graph we split x conceptually into a fractional matching z for the graph (not with the actual assigned weights, but with the help of the variables r and w). This fractional matching has size $\sum_j(\sum_{i_k:(i_k,j) \text{ exists}} z((i_k, j))) = \sum_j(\sum_i x_{ij}) = \sum_j 1 = n$ (which is the biggest fractional matching in the graph). By Fact 2, we conclude that the graph contains a matching of size n.

(f) Fix some TA i; he/she will grade the q_i exercises assigned to the nodes $i_1, \ldots, i_{q_i}$. Denote with T_c the slowest processing time of a job assigned to i_c. Then the time required will be at most $\sum_{c=1}^{q_i} T_c$. Recall that since x is a solution of the linear program with the previously assumed property, we know that $\sum_j x_{ij} = 1$ and $\sum_j x_{ij}t_{ij} \leq T(\tilde{t})$. Furthermore we have that i_c has correcting times $t_{i,j} \geq T_{c+1}$. So we can bound $\sum_{c=2}^{q_i} T_c \leq \sum_j x_{ij}t_{ij} \leq T(\tilde{t})$ with the help of the previously defined fractional matching z and the fact that all TA copies (except maybe i_{q_i}) are saturated in the matching. We left out the exercise assigned to i_q but $T_1 \leq \max_{iji} x_{i,j} \leq \tilde{t}$.

(g) Run the algorithm for each $\tilde{t} = t_{ij}$. If the LP is feasible, build the bipartite graph, find the matching and store the solution given by the matching. If the LP is not feasible, set $\tilde{t}$ to the next value and keep going. In the end, return the best approximation found. As we proved in (c), the algorithm will find a feasible program for $\tilde{t} \leq$ OPT. Finally, from (f), the cost of the solution returned will be at most $\tilde{t} + T(\tilde{t}) \leq 2$OPT.

Exercise 3 20 points

(*Random orientation*)
Let $G = (V, E)$ be a graph and let $\overrightarrow{G}$ be a graph obtained by orienting the edges of G independently with probability $1/2$ in either direction. Given $\overrightarrow{G}$, define the skew symmetric matrix A_s such that

$$A_s(i,j) = \begin{cases} +1, & (i,j) \in E \text{ with orientation from } i \text{ to } j, \\ -1, & (i,j) \in E \text{ with orientation from } j \text{ to } i, \\ 0, & \text{otherwise.} \end{cases}$$

(a) Let $\mathrm{pm}(G)$ denotes the number of *perfect matchings* in G, prove that

$$\mathbf{E}\left[\det A_s(\overrightarrow{G})\right] = \mathrm{pm}(G).$$

Hint: Use the definition of determinant and the linearity of expectation.

(b) Given two perfect matchings M_1, M_2 of G, denote with $a(M_1, M_2)$ the number of cycles in G made of (alternating) edges contained in M_1 and M_2 (or equivalently the cycles in M_1 xor M_2), and denote with $\mathcal{M}$ the set of all perfect matchings in G. Prove that

$$\mathbf{E}\left[\left(\det A_s(\overrightarrow{G})\right)^2\right] = \sum_{M_1 \in \mathcal{M}} \sum_{M_2 \in \mathcal{M}} 3^{a(M_1,M_2)}.$$

Hint: You may find inspiration in the proof of Theorem 5.3. First deal with permutations whose directed graph uses edges that do not appear in G. *Next argue about the contribution from permutations where this graph contains at least one cycle of odd length, and finally the rest.*

(c) Let

$$\mathrm{var}\left[\det A_s(\overrightarrow{G}_i)\right] = \mathbf{E}\left[\left(\det A_s(\overrightarrow{G})\right)^2\right] - \left(\mathbf{E}\left[\det A_s(\overrightarrow{G})\right]\right)^2.$$

Design a randomized algorithm that computes the determinant of at most $N = 2\,\mathrm{var}\left[\det A_s(\overrightarrow{G}_i)\right]$ matrices, runs in $O(N\ \mathrm{poly}(|V|))$ time, and returns x such that

$$\mathbf{Pr}\left[|\mathrm{pm}(G) - x| \geq 1\right] \leq \frac{1}{2}.$$

(d) Find a collection of graphs G such that $\mathrm{var}\left[\det A_s(\overrightarrow{G})\right]$, where $\overrightarrow{G}$ is a random orientation of G, is *not* polynomially bounded in the size of G, i.e. there is no polynomial p such that $\mathrm{var}\left[\det A_s(\overrightarrow{G})\right] \leq p(|V(G)| + |E(G)|)$. (This means that the algorithm we just developed does not run in polynomial time for all the input graphs.)
Hint: Consider multiple copies of a graph containing only one cycle.

A useful fact.

Theorem 3 (Chebyshev's Inequality). *Let* X *be a random variable with finite expected value* μ *and finite non-zero variance* σ^2. *Then for any real number* $k > 0$,

$$\mathbf{Pr}\left[|X - \mu| \geq k\sigma)\right] \leq \frac{1}{k^2}.$$

Answer to Exercise 3

(a)
$$\mathbf{E}\left[\det A_s(\overrightarrow{G})\right] = \mathbf{E}\left[\sum_{\pi} sign(\pi)\prod_{i} a_{i,\pi(i)}\right] = \sum_{\pi} sign(\pi)\,\mathbf{E}\left[\prod_{i} a_{i,\pi(i)}\right]$$

If there exists j such that $\pi(j) = j$ or $(j, \pi(j)) \notin E(G)$, then $\prod_i a_{i,\pi(i)} = 0$. Further, if there exists j such that $\pi(\pi(j)) \neq j$, then we have that $\mathbf{E}\left[\prod_i a_{i,\pi(i)}\right] = \mathbf{E}\left[\prod_{i\neq j} a_{i,\pi(i)}\right]\mathbf{E}\left[a_{j,\pi(j)}\right] = 0$ because $a_{j,\pi(j)}$ is independent from the other terms and has expectation 0. We are left with only the permutations π such that $\pi(\pi(i)) = i$, $\forall i$; for these, the term $\prod_i a_{i,\pi(i)} = 1 = (\pi)$ because each variable appears exactly 2 times in the sum and they consist of even cycles. Finally it is not hard to see that there exists a bijection between these permutations and the perfect matchings of G. This concludes the argument.

(b) We first plug in the definition of the determinant

$$\mathbf{E}\left[\left(\det A_s(\overrightarrow{G})\right)^2\right] = \mathbf{E}\left[\left(\sum_{\pi} sign(\pi)\prod_{i} a_{i,\pi(i)}\right)^2\right].$$

We partition the set of permutations into a set

$$A := \{\pi : \exists j : \pi(j) = j \vee (j, \pi(j)) \notin E(G)\}$$

and its complement

$$B := \{\pi : \pi \notin A\}.$$

Every permutation in A contributes a zero term to the product, and therefore we have

$$\begin{aligned}
\mathbf{E}\left[\left(\sum_{\pi} sign(\pi)\prod_{i} a_{i,\pi(i)}\right)^2\right] &= \mathbf{E}\left[\left(\sum_{\pi\in B} sign(\pi)\prod_{i} a_{i,\pi(i)}\right)^2\right] \\
&= \mathbf{E}\left[\sum_{\pi_1\in B}\sum_{\pi_2\in B} sign(\pi_1)\cdot sign(\pi_2)\prod_{i} a_{i,\pi_1(i)}\prod_{i} a_{i,\pi_2(i)}\right] \\
&= \mathbf{E}\left[\sum_{\pi_1\in B}\sum_{\pi_2\in B} val(\pi_1)\cdot val(\pi_2)\right]
\end{aligned}$$

where $val(\pi) := sign(\pi)\prod_i a_{i,\pi(i)}$. We then define another set of permutation $C \subset B$:

$$C := \{\pi \in B : \text{all cycles in } \pi \text{ are of even length}\}$$

. Next, we show that

$$\mathbf{E}\left[\sum_{\pi_1\in B}\sum_{\pi_2\in B} val(\pi_1)\cdot val(\pi_2)\right] = \mathbf{E}\left[\sum_{\pi_1\in C}\sum_{\pi_2\in C} val(\pi_1)\cdot val(\pi_2)\right]. \qquad (1)$$

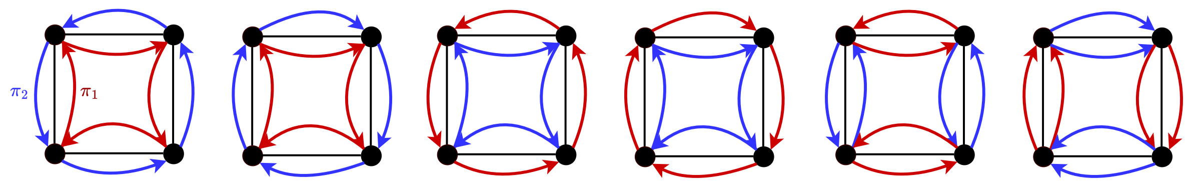

Figure 1: The six options of permutations on an (even) cycle. Notice that the same permutation types exist on a longer (even) cycle. The red arrows go from vertex i to vertex $\pi_1(i)$ while the blue arrows go from vertex i to vertex $\pi_2(i)$.

To do so, we fix some permutation $\pi_1 \in B$ that contains an odd cycle and an arbitrary permutation $\pi_2 \in B$. Then consider the odd cycle c incident to the lowest indexed vertex among the odd cycles in π_1. If this odd cycle does not appear in π_2 in some direction, there is a term a_{ij} that only appears once in $val(\pi_1) \cdot val(\pi_2)$. Therefore $\mathbf{E}[val(\pi_1) \cdot val(\pi_2)] = 0$ because the edge directions are chosen independently. Similarly, this holds for every (odd) cycle in π_1. Therefore the cycle c is also incident to the lowest indexed vertex on an odd cycle in π_2. We now pair permutations 1: (π_1, π_2), 2: $(\pi_1' = \pi_1$ with c reversed, $\pi_2)$, 3: $(\pi_1,\ \pi_2' = \pi_2$ with c reversed) and 4: $(\pi_1',\ \pi_2')$. Since

$$val(\pi_1) \cdot val(\pi_2) = -val(\pi_1') \cdot val(\pi_2) = -val(\pi_1) \cdot val(\pi_2') = val(\pi_1') \cdot val(\pi_2')$$

these 4 cancel out and this establishes (1).

Finally, we show that

$$\mathbf{E}\left[\sum_{\pi_1 \in C} \sum_{\pi_2 \in C} val(\pi_1) \cdot val(\pi_2)\right] = \sum_{M_1 \in \mathcal{M}} \sum_{M_2 \in \mathcal{M}} 3^{a(M_1, M_2)}. \tag{2}$$

We fix two matchings M_1 and M_2 and we let the support E_s denote the set of edges that are either in M_1, in M_2 or in both. This support consists of even cycles and isolated edges. Given a fixed support E_s, we are interested in the amount of (ordered) matching pairs that yield E_s. For every isolated edge there is exactly one option: Both M_1 and M_2 have to contain it. But for each cycle there are exactly two possible options. Therefore, the amount of matchings is given by the $2^{\#\text{even cycles}}$. Now, we consider the amount of pairs of permutations π_1, π_2 that live on this support. For each isolated edge (i, j), there is exactly one choice for both π_1 and π_2. For each even cycle however, there are 6 choices. These are displayed in Figure 1. Therefore, the total number of permutations supported is $6^{\#\text{even cycles}}$, and there is a mapping from each pair of matchings yielding the support to $3^{\#\text{even cycles}}$ permutations. In such a way, each permutation is allocated to a pair of matchings. This establishes (2) and concludes the solution to this exercise.

(c) Sample N independent random orientations for the edges of G and let $\overrightarrow{G}_1, \ldots, \overrightarrow{G}_N$ be these graph. The algorithm returns $\frac{1}{N}\sum_{i=1}^{N} \det A_s(\overrightarrow{G}_i)$. The variance of this quantity is given by $1/N$ times the variance of an individual run by independence of the runs and $var[X+Y] = \text{var}[[]X] + \text{var}[[]Y]$ for X and Y independent, and the fact that $var[c \cdot X] =$

$c^2 \cdot var[X]$. That is

$$\sqrt{\mathrm{var}\left[\frac{1}{N}\sum_{i=1}^{N}\det A_s(\overrightarrow{G}_i)\right]} = \frac{1}{\sqrt{N}}\sqrt{\mathrm{var}\left[\det A_s(\overrightarrow{G}_i)\right]}.$$

By Chebyshev's inequality:

$$\mathbf{Pr}\left[\left|\mathbf{E}\left[\det A_s(\overrightarrow{G})\right] - \frac{1}{N}\sum_{i=1}^{N}\det A_s(\overrightarrow{G}_i)\right| \geq k\sqrt{\mathrm{var}\left[\frac{1}{N}\sum_{i=1}^{N}\det A_s(\overrightarrow{G}_i)\right]}\right] \leq \frac{1}{k^2}$$

For $k = \sqrt{2}$, we obtain the desired bound. Each iteration of the algorithm computes a determinant. Using gaussian elimination, this can be implemented in polynomial time.

(d) Fix $n \in \mathbb{N}$, divisible by 4 and consider the graph G made by the union of $n/4$ copies of the cycle graph on 4 edges C_4.
Since C_4 has two perfect matchings, we have that

$$pm(G) = 2^{n/4}.$$

Consider the pairs of matchings $M_1, M_2 \in \mathcal{M}$ that are disjoint (i.e. they do not share any edge). There are exactly $2^{n/4}$ such pairs because for each copy of C_4 we have 2 possible combinations. Furthermore, if M_1 and M_2 are disjoint, the graph obtained by the *xor* of the edges is G and $a(M_1, M_2) = n/4$. Using the results from the previous part of the exercise,

$$\begin{aligned}
\mathrm{var}\left[\det A_s(\overrightarrow{G}_i)\right] &= \mathbf{E}\left[\left(\det A_s(\overrightarrow{G})\right)^2\right] - \left(\mathbf{E}\left[\det A_s(\overrightarrow{G})\right]\right)^2 \\
&= \sum_{M_1\in\mathcal{M}}\sum_{M_2\in\mathcal{M}} 3^{a(M_1,M_2)} - pm(G)^2 \\
&\geq \sum_{M_1,M_2 \text{ disjoint}} 3^{a(M_1,M_2)} - pm(G)^2 \\
&= 2^{n/4}\, 3^{n/4} - 2^{n/2} \\
&\geq \frac{3}{2}2^{n/2} - 2^{n/2} = \frac{1}{2}2^{n/2}
\end{aligned}$$