

Example exam in AADS 2024–2025

Københavns Universitet

Instructions

You have four hours to complete the exam. You are allowed to use aids but you are not allowed to use your own computers, tablets, calculators, mobile phones or headphones.

The exam has two parts. A “multiple-choice part” and an “open part”. There are 12 multiple-choice questions and 4 open questions. Each of the 12 multiple choice question is worth 6 points. Each of the open questions is worth 7 points. The total number of points is 100.

Multiple-choice part. You should start your submission with your solutions to the multiple-choice part. Your submission should start with 12 rows. And on row i you should write i and then your answers in numbers. As an example, it should start in this format:

1. 2, 3, 5
2. 2
3. 5, 6
- \vdots
12. 5

In all multiple-choice questions, there could be more than one true option. To get a full grade (6 points), you should mark exactly all the correct options. If you mark a non-empty subset of the correct options you get half the points (3 points). In any other case, you get 0 points.

Open part. The solutions to the open questions should appear in the order of the questions. The solutions to question i should start with the title “Solution to question i ”. Write your solutions clearly and accurately. Your grade will be determined both by the correctness and by the clarity of your answers.

Notation. Throughout the exam we use $[n]$ to denote the set $[n] = \{1, 2, \dots, n\}$.

I. Multiple-choice questions

1 Random cuts (6%)

Let $n > 3$ be an integer and let G_n be the n -cycle graph with one edge extra attached to it. That is, its vertex set is $[n + 1]$ and its edges are $\{1, 2\}, \{2, 3\}, \dots, \{n - 1, n\}, \{n, 1\}$ and $\{1, n + 1\}$. Run the randomized min-cut algorithm for one iteration on G_n (that is, just until the algorithm finds a cut—this cut could be a non-min cut). What is the probability p_n that algorithm outputs a minimum cut?

1. $p_n \geq \Omega(\frac{1}{n})$.
2. $p_n \geq \Omega(\frac{1}{n^2})$.
3. $p_n \geq \Omega(1)$.
4. $p_n \leq O(\frac{1}{n})$.
5. $p_n \leq O(\frac{1}{n^2})$.
6. $p_n \leq O(1)$.

2 Hashing (6%)

Let $n > 10$ be an integer. Let H be a strong universal hash function from $[n]$ to $[n]$. That is, $\Pr[H(x) = a, H(y) = b] = \frac{1}{n^2}$ for all $a, b \in [n]$ and all distinct $x, y \in [n]$. Let C be the number of collisions:

$$C = \{(x, y) \in [n] \times [n] : H(x) = H(y)\}.$$

(It is a random variable.) What is the expected value of C ?

1. Exactly n .
2. Exactly $n - 1$.
3. At most $10n$.
4. At most $\frac{n}{2}$.
5. At most \sqrt{n} .

3 General complexity (6%)

Which of the following is true?

1. There is a function $f : \{0,1\}^* \rightarrow \{0,1\}$ that can not be computed by a Turing machine.
2. There is an integer n and there is a function $f : \{0,1\}^n \rightarrow \{0,1\}$ so that f can not be computed by a Turing machine.
3. If $P = NP$, then SAT can be decided in time $O(2^{\sqrt{n}})$.
4. If $P = NP$, then there is an NP-hard language L so that $L \notin P$.

4 Small languages (6%)

We say that a language $L \subset \{0, 1\}^*$ is small if for every integer $n > 1$, we have that $|L \cap \{0, 1\}^n| \leq n^2$. Which of the following is true?

1. There is a small language L that can not be decided by a Turing machine.
2. There is a small language L that is not in P .
3. There is a small language L in P .
4. If L is small and $L \in P$ then the language $\{(1^n, k) : |L \cap \{0, 1\}^n| = k\}$ can be decided by a Turing machine.
5. If L is not small, then L is not in P .
6. If L is small, then L is in P .

5 Linear programming (6%)

Let

$$A = \begin{bmatrix} 2 & 1 \\ -5 & 1 \\ 0 & -1 \\ 1 & 1 \end{bmatrix} \quad \text{and} \quad b = (5, 5, 0, 6).$$

Let $P = \{x \in \mathbb{R}^2 : Ax \leq b\}$. Which of the following is true?

1. The polytope P has 4 vertices.
2. Let $F(x) = \langle c, x \rangle$ where $c = (1, 1)$. Then, $\max\{F(x) : x \in P\} = 5$.
3. Let $G(x) = \langle c, x \rangle$ where $c = (-10, 2)$. There is $x \in P$ so that for all $x' \in P$ so that $x' \neq x$ we have $G(x') < G(x)$.
4. Let $Q = \{y \in \mathbb{R}^4 : A^T y \geq c\}$ where $c = (1, 1)$, and let $H : \mathbb{R}^4 \rightarrow \mathbb{R}$ be defined by $H(y) = \langle b, y \rangle$. Then, $\min\{H(y) : y \in Q\} \geq 6$.

6 Art gallery (6%)

In this question, the guards can guard in a 90° angle (if the guard is placed at the point $x \in \mathbb{R}^2$, there are two perpendicular lines through x not necessarily parallel to the axis and the guarded area is one of the quadrants defined by the lines). What is the minimum number k of 90° -angle guards that are needed to guard a convex polygon $P \subset \mathbb{R}^2$?

1. $k = 1$
2. $k = 2$
3. $k = 3$
4. We need more information about P to decide k .

7 Helpers (6%)

We apply the algorithm `MAKEMONOTONE` from page 53 of Chapter 3 of the book *Computational Geometry: Algorithms and Applications* to the polygon in Figure 1. What are the helpers of the edges bc , ef , hi just after the vertex e has been handled?

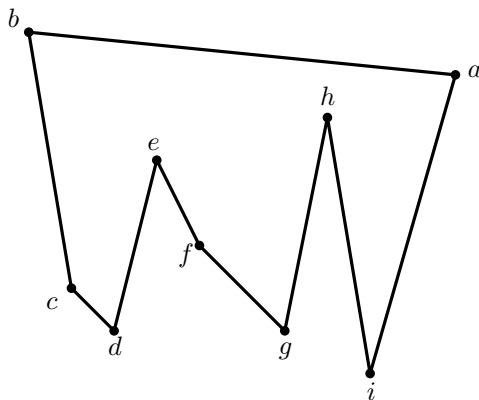


Figure 1: A polygon

1. $\text{helper}(bc) = b, \text{helper}(ef) = e, \text{helper}(hi) = h$
2. $\text{helper}(bc) = h, \text{helper}(ef) = h, \text{helper}(hi) = a$
3. $\text{helper}(bc) = c, \text{helper}(ef) = f, \text{helper}(hi) = i$
4. $\text{helper}(bc) = e, \text{helper}(ef) = e, \text{helper}(hi) = h$

8 Approximate covers (6%)

The SET-COVER problem generalizes the VERTEX-COVER problem. Every input to VERTEX-COVER is also an input to SET-COVER. In class, we have seen a (matching-based) approximation algorithm APPROX-VERTEX-COVER for vertex cover and a greedy approximation algorithm GREEDY-SET-COVER for set cover.

Let $G = (V, E)$ be an input to VERTEX-COVER. Which of the following is true?

1. We can reduce the problem of finding a minimum vertex cover in G to an instance of set cover $I = (X, \mathcal{F})$ where X is E and there is one set in \mathcal{F} for each $v \in V$.
2. We always get a better approximation ratio by using the algorithm GREEDY-SET-COVER on the instance G instead of using the approximation algorithm APPROX-VERTEX-COVER for vertex cover directly on G .
3. There are graphs on which using GREEDY-SET-COVER on the instance I performs worse than applying APPROX-VERTEX-COVER directly on G .
4. The algorithm APPROX-VERTEX-COVER does not necessarily find a matching in G of the maximum possible size.
5. The algorithm APPROX-VERTEX-COVER for vertex cover always finds a matching in G of the maximum possible size.

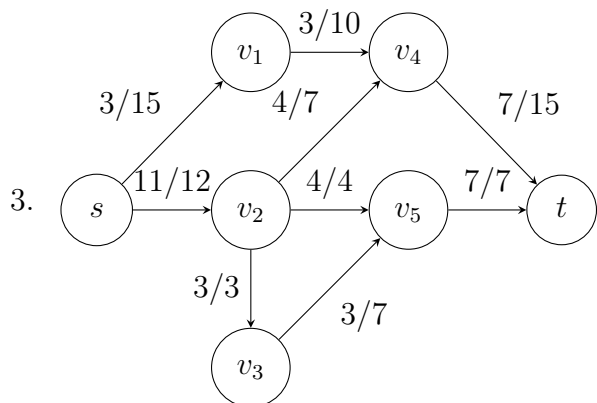
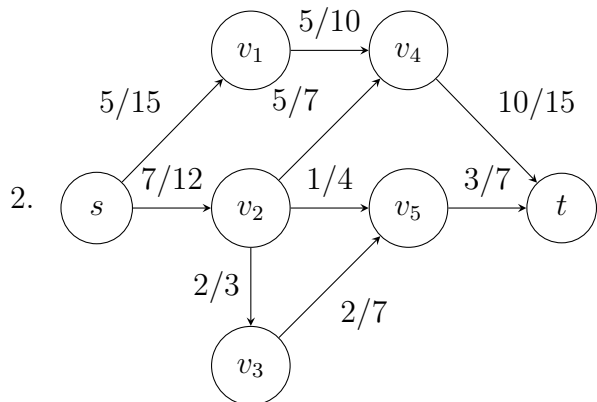
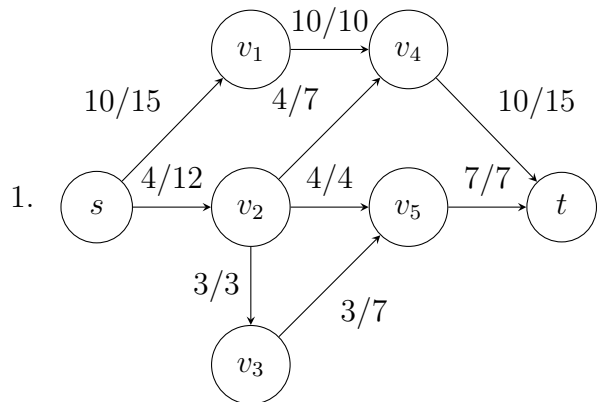
9 FPTAS (6%)

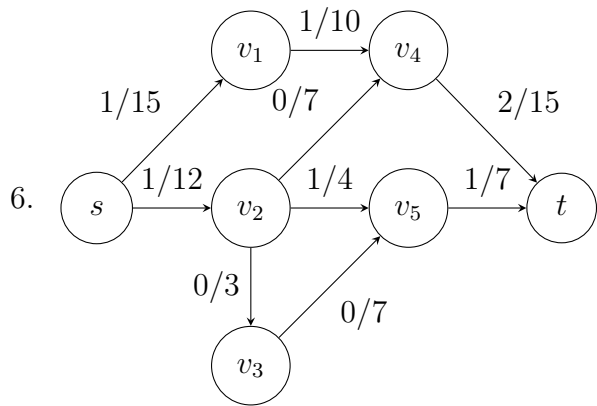
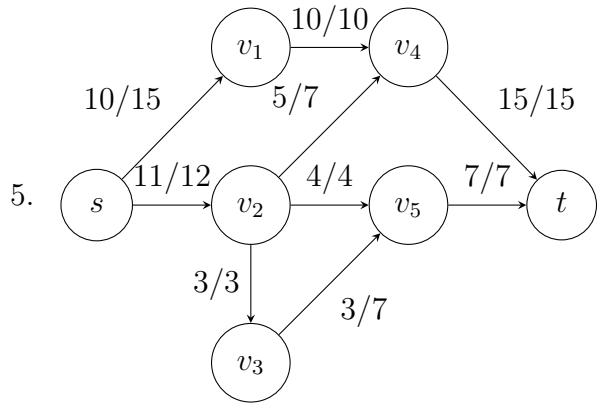
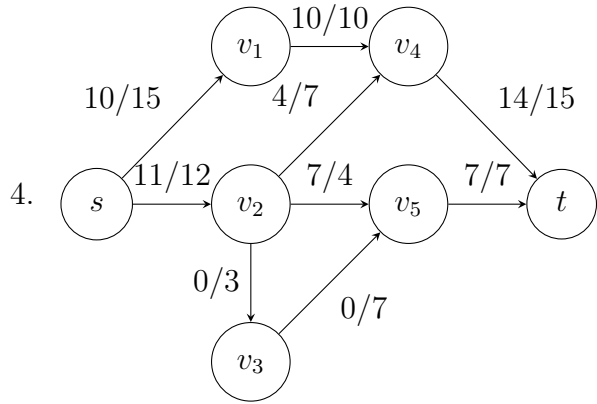
For some optimization problem, we have a PTAS which achieves an approximation ratio of $1 + \varepsilon$ in $f(n, \varepsilon)$ time for any fixed $\varepsilon > 0$. For which of the following cases is it an FPTAS?

1. $f(n, \varepsilon) = n/\varepsilon$.
2. $f(n, \varepsilon) = (1/\varepsilon)^n$.
3. $f(n, \varepsilon) = (1/\varepsilon)^{\log n}$.
4. $f(n, \varepsilon) = (\log n)^{1/\varepsilon}$.
5. $f(n, \varepsilon) = 2^{\log n} + 2^{\log(1/\varepsilon)}$.
6. $f(n, \varepsilon) = 100^{\log n} \cdot 2^{100 \log(1/\varepsilon)}$.

10 Valid flows (6%)

Which of the following examples of flows in a flow network are valid?





11 Max flow (6%)

In this course we defined the flow value of a valid flow f in $G = (V, E)$ from a source s to a sink t as the amount of flow going out of the source minus the amount of flow going into the source:

$$V_s(f) = \left(\sum_{v \in V} f(s, v) \right) - \left(\sum_{v \in V} f(v, s) \right)$$

Consider the case where we define it as the amount of flow going into the sink minus the amount of flow going out of the sink:

$$V_t(f) = \left(\sum_{v \in V} f(v, t) \right) - \left(\sum_{v \in V} f(t, v) \right).$$

Are the two definitions the equivalent (that is, $V_s(f) = V_t(f)$) and why?

1. They are not equivalent because of the capacity constraint.
2. They are equivalent because of flow conservation.
3. They are not equivalent because of flow conservation.
4. They are equivalent because of the max-flow min-cut theorem.
5. They are equivalent because of the capacity constraint.
6. They are not equivalent because of the max-flow min-cut theorem.

12 TSP algorithm (6%)

We showed in class that there is a polynomial-time 2-approximation algorithm for the TSP problem when the triangle inequality holds. Which of the following statements is true for an instance (G, c) of TSP when the triangle inequality is *not* satisfied?

1. Let T be any spanning tree of G . Then the cost of T is always at most the cost of the shortest TSP tour H^* .
2. Let T be any minimum spanning tree of G . Then the cost of T is always at most the cost of the shortest TSP tour H^* .
3. It may be the case that every spanning tree of G has larger cost than the cost of H^* .
4. It may be the case that the cost of the full-walk W on T (as defined in the algorithm) has larger cost than twice the cost of T .
5. It may be the case that the cost of the tour H constructed by the algorithm from the full-walk W is larger than the cost of W .

II. Open questions

1 Flows (7%)

(a) Define the notion “augmenting path”.

(b) Let G be a flow network with a source s and a sink t . Denote by n the number of vertices in G . Assume that the number of edges on the shortest path from s to t is D . Also assume that for every node $v \notin \{s, t\}$, there is a unique path from s to v . Assume that all capacities are in $[0, 1]$. Prove that the maximum flow value on G is at most $\frac{2n}{D}$.

2 Randomized algorithms (7%)

(a) If we have a coin that lands on “head” with chance $p \in (0, 1)$, what is the expected number of time we need to toss it until we get the first “head”?

(b) Let $G = (V, E)$ be a graph with $|V| = 300$ vertices and $|E| = m$ edges. For $S \subseteq V$, denote by $e(S, S^c)$ the number of edges in G with one endpoint in S and one endpoint outside S . Prove that there is $S \subseteq V$ of size $|S| \leq 100$ so that $e(S, S^c) \geq \frac{4m}{9}$.

3 Complexity (7%)

(a) Define the notion “the language L is NP-complete”.

(b) Assume that $P = NP$. Show that there is a polynomial-time algorithm that takes as input a graph $G = (V, E)$, and outputs a set of vertices $K \subseteq G$ that forms a clique in G of maximum size.

4 Approximation (7%)

(a) Consider a term of the form $\ell_1 \vee \ell_2 \vee \ell_3 \vee \ell_4$ where ℓ_1, \dots, ℓ_4 are literals. If the input to the term is chosen uniformly at random, what is the chance that the term evaluates to 1?

(b) If we run the RANDOM-ASSIGNMENT algorithm for MAX-4-SAT instead of MAX-3-SAT, is it still an $\frac{8}{7}$ -approximation? If your answer is yes, prove it. If your answer is no, give a new approximation ratio and prove it.