

University of Waterloo
CO 353 Midterm Examination

Winter 2018

Solutions

Exam details:

Course:	CO 353 – Computational discrete optimization
Instructor:	Laurent Poirrier
Date and time of exam:	February 16, 2018 – 11:30 AM to 12:20 PM
Duration of exam:	50 minutes
Number of exam pages:	4 (includes this cover page)
Exam type:	Closed book – no additional material allowed

Instructions:

Your answers must be stated and justified in a clear and logical form, and you must show all of your steps in order to receive full marks. You may use any result from class without proof, unless you are being asked to prove this result. You will be graded not only on correctness, but also on clarity of exposition. No collaboration is allowed.

Question 1

a. [2 marks] Consider the following problems. For each of them, give the encoding size, in big-O notation. No justification is necessary for this subquestion.

1. Given $a, b \in \mathbb{Z}_+$, compute

$$a^b \tag{P1}$$

Solution:

$$O(\log a + \log b)$$

2. Given $a \in \mathbb{Z}_+^2, b \in \mathbb{Z}_+, c \in \mathbb{Z}_+^2, u \in \mathbb{Z}_+^2$, compute

$$\begin{array}{ll} \max & c_1x_1 + c_2x_2 \\ \text{s.t.} & a_1x_1 + a_2x_2 = b \\ & 0 \leq x_1 \leq u \\ & 0 \leq x_2 \leq u \\ & x_1, x_2 \in \mathbb{Z}, \end{array} \tag{P2}$$

Solution:

$$O(\log(c_1) + \log(c_2) + \log(a_1) + \log(a_2) + \log(b) + \log(u_1) + \log(u_2))$$

- b. [2 marks] Consider the following algorithms. For each of them, give its computational complexity, in big-O notation, in the *arithmetic* model. No justification is necessary for this subquestion.

1. Given $a, b \in \mathbb{Z}_+$,

```

 $z := 1$ 
while  $b > 0$ 
|   if  $b$  is odd
|        $z := az$ 
|        $b := b - 1$ 
|    $b := b/2$ 
|    $a := a^2$ 
return  $z$ 

```

(A1)

Solution:

$$O(\log(b))$$

2. Given $a \in \mathbb{Z}_+^2, b \in \mathbb{Z}_+, c \in \mathbb{Z}_+^2, u \in \mathbb{Z}_+^2$,

```

 $z := -1$ 
for  $x_1 = 0, 1, \dots, u_1$ 
|   for  $x_2 = 0, 1, \dots, u_2$ 
|       if  $a_1x_1 + a_2x_2 = b$ 
|            $w := c_1x_1 + c_2x_2$ 
|           if  $w > z$ 
|                $z := w$ 
return  $z$ 

```

(A2)

Solution:

$$O(u_1 \cdot u_2)$$

- c. [1 mark] Let L be the encoding size of problem (P2) above, and assume that (A2) above is an algorithm that solves (P2). Consider the complexity of (A2) in the arithmetic model. Is it polynomial in L ? If yes, give the polynomial in L , in big-O notation. If not, explain why it is not polynomial in L .

Solution:

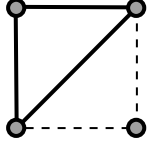
We have $L = O(\log(c_1) + \log(c_2) + \log(a_1) + \log(a_2) + \log(b) + \log(u_1) + \log(u_2))$. The complexity is $C = O(u_1 \cdot u_2)$, and it is not polynomial in L because L only has terms in $\log(u_1)$ and $\log(u_2)$: there exist no polynomial in L that can express u_1 or u_2 (instead, we could express for example u_1 as an exponential in L : $u_1 = O(2^L)$).

Question 2 [3 marks] Prove that the statement below is false by giving a counter-example. Draw a counter-example graph G , a subgraph H (use colors or symbols or dashed lines to differentiate G and H in your drawing), and state why the statement is false in the counter-example.

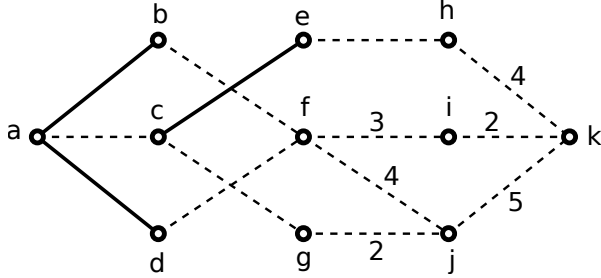
Wrong Theorem: Let $G(V, E)$ be a connected graph and $H(W, F)$ be a subgraph of G . If $|F| = |V| - 1$ then H is a spanning tree of G .

Solution:

Consider the following graph $G(V, E)$ (continuous and dashed edges) and its subgraph $H(W, F)$ (continuous edges only). The subgraph H has 3 edges, which is indeed $|V| - 1$, and yet it is not spanning. Note that any subgraph H with a circuit could be a counter-example.



Question 3 [6 marks] Consider the graph $G(V, E)$ illustrated below (continuous and dashed edges: $V = \{a, b, c, \dots, j, k\}$, $E = \{ab, ac, ad, bf, ce, cg, df, eh, fi, fj, gj, hk, ik, jk\}$), and its subgraph $H(W, F)$ (continuous edges only: $W = \{a, b, c, d, e\}$, $F = \{ab, ad, ce\}$). The edge costs $c_e > 0$ are given for some edges ($c_{fi} = 3, c_{fj} = 4, c_{gj} = 2, c_{hk} = 4, c_{ik} = 2, c_{jk} = 5$), but are unknown for the others. Assuming that H can be extended to a minimum spanning tree of G (i.e., H is a subgraph of a minimum spanning tree of G), prove that $H'(W \cup \{f, i\}, F \cup \{fi\})$ can also be extended to a minimum spanning tree of G . You can use without proof any result seen in class.



Solution:

We use the following lemma seen in class: Let H be a subgraph of G that can be extended to an MST. Let D be a cut in G such that $H \cap D = \emptyset$. Let e be a minimum-cost edge in D . Then, $H \cup \{e\}$ can be extended to an MST.

In this case, consider the cut $D = \delta(\{i, k\}) = \{hk, fi, jk\}$. The costs of all edges in D are known (4, 3, and 5, respectively), and fi 's cost is minimum (3). Hence, the lemma applies and H' can also be extended to an MST.

Question 4 [6 marks] Given, $a \in \mathbb{Z}_+^n$ and $b \in \mathbb{Z}_+$, consider the constraints of a 0-1 knapsack:

$$\begin{aligned} \sum_{j=1}^n a_j x_j &\leq b \\ x_j &\in \{0, 1\}, \quad \forall j = 1, 2, \dots, n. \end{aligned} \tag{K}$$

Let $S := \{1, 2, \dots, n\}$ and $\mathcal{I} := \{J \subseteq S : \sum_{j \in J} a_j \leq b\}$. Note that \mathcal{I} corresponds to the set of all feasible solutions to (K), where $j \in J$ if we take the j th object in the knapsack. Is (S, \mathcal{I}) a matroid? If yes, prove that it satisfies the conditions for being a matroid. If not, give an example (S, \mathcal{I}) instance and show that it does not satisfy one of these conditions.

Solution:

(S, \mathcal{I}) is not a matroid. Consider the 0-1 knapsack

$$x_1, x_2, x_3 \in \{0, 1\} : x_1 + x_2 + 2x_3 \leq 2,$$

with ground set $S = \{1, 2, 3\}$. Then, let $A = S = \{1, 2, 3\}$. Clearly, $\{1, 2\}$ is a basis of A ($1 + 1 + 0 \leq 2$ but $1 + 1 + 2 \not\leq 2$), and $\{3\}$ is another one ($0 + 0 + 2 \leq 2$ but $0 + 1 + 2 \not\leq 2$ and $1 + 0 + 2 \not\leq 2$). We can observe that $|\{1, 2\}| = 2 \neq |\{3\}| = 1$, so not all bases of A have the same cardinality. Therefore, (S, \mathcal{I}) is not a matroid.

Question 5 [5 marks] Consider two problems H and Q , both in \mathcal{NP} . For each statement below, state whether it is true or false. No justification is necessary for this question.

1. If H is in \mathcal{P} and there is a polynomial reduction from Q to H , then this implies that Q is in \mathcal{P} .

Solution:

True.

2. If Q is in \mathcal{P} and there is a polynomial reduction from Q to H , then this implies that H is in \mathcal{P} .

Solution:

False. It only implies that Q is not much harder than H , but H could still be harder than Q .

3. If there is a polynomial reduction from H to Q then this implies that Q is \mathcal{NP} -complete.

Solution:

False. By definition, Q is \mathcal{NP} -complete if there is a polynomial reduction from **all** problems in \mathcal{NP} to Q , not just H . The statement would be true if H was known to be \mathcal{NP} -complete.

4. If there is a polynomial reduction from Q to H then this implies that Q is \mathcal{NP} -complete.

Solution:

False. Consider the two cases where the complexity of H can tell us something about Q . Either $H \in \mathcal{P}$, then it only implies that $Q \in \mathcal{P}$. Or H is \mathcal{NP} -complete, then there is always a reduction from Q to H , whether $Q \in \mathcal{P}$ or not. In any case, nothing can be said about Q being \mathcal{NP} -complete.

5. If (i) H is \mathcal{NP} -complete and (ii) Q is in \mathcal{P} and (iii) there is a polynomial reduction from Q to H , then it implies that $\mathcal{P} = \mathcal{NP}$.

Solution:

False. If (i) and (ii) hold, then (iii) always also holds, whether or not $\mathcal{P} \neq \mathcal{NP}$. Statement would be true if (iii) was from H to Q .