CPSC 320 Sample Midterm 1
February, 2016

Name: ______________________ Student ID: ______________________

Signature: ______________________

- You have 70 minutes to write the 5 questions on this examination. A total of 70 marks are available.
- **Justify all of your answers.**
- You are allowed to bring in one hand-written or printed, double-sided 8.5 x 11in sheet of notes, and nothing else.
- Keep your answers short. If you run out of space for a question, you have written too much.
- The number in square brackets to the left of the question number indicates the number of marks allocated for that question. Use these to help you determine how much time you should spend on each question.
- Use the back of the pages for your rough work.
- **Good luck!**

Question	Points	Score
1	20	
2	8	
3	24	
4	10	
5	8	
Total:	70	

UNIVERSITY REGULATIONS:

- Each examination candidate must be prepared to produce, upon the request, his/her UBC card.
- No examination candidate shall be permitted to enter the examination room after the expiration of one half hour, or to leave during the first half hour of the examination.
- CAUTION: candidates guilty of any of the following, or similar, dishonest practices shall be immediately dismissed from the examination and shall be liable to disciplinary action.
  1. Having at the place of writing, or making use of, any books, papers or memoranda, electronic equipment, or other memory aid or communication devices, other than those authorised by the examiners.
  2. Speaking or communicating with other candidates.
  3. Purposely exposing written papers to the view of other candidates. The plea of accident or forgetfulness shall not be received.
- Candidates must not destroy or mutilate any examination material; must hand in all examination papers; and must not take any examination material from the examination room without permission of the invigilator.

1. [20 points] Short Answers

(a) [4 points] Consider the Gale-Shapley stable matching algorithm. Is it possible for every woman to get the worst partner on her list? If so, what preference lists would result in such outcome?

(b) [8 points] Rank the following functions by order of growth. Use $f \ll g$ if $f \in o(g)$ and $f \approx g$ if $f \in \Theta(g)$.

$$n \log n \qquad \log^3 n \qquad 2^{\log_2 n} \qquad n^2 / \log n$$

You **don't** need to justify your answer.

(c) [4 points] Let P be a shortest path from s to t in a directed graph $G = (V, E)$ with distance function $\ell : E \to \mathbb{R}^+$. Let $\ell' : E \to \mathbb{R}^+$ is defined as follows: $\ell'(e) = [\ell(e)]^2$ for every $e \in E$. **Decide whether the following statement is true or false (justify your answer shortly).**

Statement: *P is always a shortest path from s to t in G with distance function ℓ'.*

(d) [4 points] In the decision tree build for an algorithm $\mathcal{A}$ solving a problem $\mathcal{P}$ on inputs of size n,

(i) what is the number of children (outgoing edges) of each internal node in the tree?

(ii) what can we say about the number of leaves of the tree with respect to the problem $\mathcal{P}$?

---

2. [8 points] Consider three functions $f, g, h : \mathbb{N} \to \mathbb{R}^+$, where $f(n) \in \Omega(g(n))$.

   What can we say about asymptotic relation between $f(n)h(n)$ and $g(n)h(n)$? **Justify** your answer (showing an example of functions that satisfy this relation does not count).

---

3. [24 points] Interval Scheduling.

   (a) [6 points] Consider interval scheduling problem from the class (selecting the maximum number of compatible intervals from set $\mathcal{I}$). Alex has proposed a greedy algorithm (different from the one considered on the lecture) that attempts to solve this problem. In attempt to prove correctness of his algorithm, Alex proved the following claim:

   Claim. Let $\mathcal{G} = \{I_1, \ldots, I_k\}$ be a greedy solution produced by Alex's algorithm and $\mathcal{O} = \{J_1, \ldots, J_m\}$ with $k \le m$ be an optimal solution. Both sequences are sorted by the starting times. Then

   - for every $x = 1, \ldots, k$, $s(I_x) \le s(J_x)$;
   - $\mathcal{G}$ is a compatible subset of $\mathcal{I}$; and
   - every interval in $\mathcal{I}$, overlaps at least one interval in $\mathcal{G}$.

   Can this claim be used to show that $\mathcal{G}$ is optimal? **Either prove it or give a counterexample.**

(b) [12 points] Consider the following interval scheduling problem:

> Your boss has just given you a list of n jobs to perform. Each job J has a starting time $s(J)$ and finishing time $f(J)$. You can never do more than one job at a time. You are tired, so you would like to do as few jobs as possible, but cannot just do "nothing". More precisely, you want to find a list of the smallest number of jobs you can do so that every **other** jobs overlaps with at least one job on your list (so the boss cannot say that you can add another job to your list).

Propose a reasonably efficient greedy algorithm. Your algorithm **does not need** to always return the smallest list, but it should make a good attempt it. Write a pseudocode for your algorithm.

(c) [6 points] **Analyze the time complexity** (find a good asymptotic upper bound on the *worst-case* running time) of your algorithm from part (b). Specify only as much of your implementation (for instance, data structures used) as needed for the analysis. Justify your answer!

Remarks: You don't need to show that your bound is tight. Remember you are not going to be judged by how low your upper bound is, but by whether it's correct and by how good it is. For instance, if your algorithm requires time $O(n^3)$, but you will claim its running time is in $O(n \log n)$, you will most likely get 0 points. But also if it works in time $\Theta(n \log n)$, and your bound is $O(n^3)$, you will not get many points either.

---

4. [10 points] **Optimal caching.**

   (a) [4 points] In case 3 of the proof of the theorem that we later used to show that the greedy caching strategy is optimal, we assumed that S_{FF} evicts an element e from the cache, whereas S_j ejects a different element f. We then stated that one of three things **must** happen before a request for e comes along:

   - A request for f, where S_j evicts e.
   - A request for f, where S_j evicts an element other than e.
   - A request for some other element x, where S_j evicts e.

   Why must one of these situations occur before any request for e?

   (b) [6 points] Consider the following online algorithm for caching:

   Least-Recently-Used (LRU): Evict the least recently used item from the cache, when there is a cache miss.

   Design an initial state of the cache and a sequence of requests that will require an eviction for each request if we follow the LRU algorithm. Justify why your answer is correct.

5. [8 points] (a) [2 points] Explain what's the different between a walk and a path in a graph.

(b) [6 points] Analyze the running time of the following algorithm:

function DIJKSTRA($G = (V, E), \ell, s$)
    $S \leftarrow \{s\}$ ▷ explored nodes
    $d(s) \leftarrow 0$ ▷ distance from s
    while $S \neq V$ **do**
        for all $v \in V \setminus S$ **do**
            $d(v) \leftarrow \min_{u \in S\colon (u,v) \in E} [d(u) + \ell(u, v)]$
            $P(v) \leftarrow \arg\min_{u \in S\colon (u,v) \in E} [d(u) + \ell(u, v)]$
        end for
        select $u \in V \setminus S$ with the minimum $d(u)$
        add u to S
    end while
    return d, P
end function

in terms of n (the number of nodes) and m (the number of edges). Find a tight asymptotic upper bound.

End of exam.