CPSC 320 Midterm Examination 1
Wed Oct 9, 2024, 09:00-09:45

Name: ____________________ Student ID: ____________________

CWL: ____________________ Signature: ____________________

- You have 45 minutes to write the three questions on this examination.
- **Please write answers in pen (preferred) or VERY DARKLY in pencil.**
- You are allowed one double-sided sheet of notes, but no laptop, calculator or other electronic equipment.
- Scrap paper for rough work is available upon request, but **you must write all your answers in the exam booklet** (we will not accept scrap paper for grading). If you run out of space to answer a question, there is extra space at the end of the exam.
- Use the number of marks allocated for each question to help you determine how much time you should spend on it. There are a total of 25 points.

Question	1	2	3
Points	6	10	9

- **Good luck!**

UNIVERSITY REGULATIONS:

- Candidates should be prepared to produce, upon request, their UBC card.
- 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.

Write CWL here (we will deduct a mark if this is incorrect or missing):

1 SMP, Asymptotic Analysis

No justification is needed for these questions.

1. [2 points] Complete the preference lists for three employers and three applicants, such that the Gale-Shapley algorithm (provided on page 7) matches employer e_3 with applicant a_3.

e_1:	a_2		
e_2:	a_2		
e_3:	a_1	a_2	a_3

a_1:	e_1		
a_2:	e_1		
a_3:	e_1	e_2	e_3

2. [2 points] Let $f(n)$ be the number of different SMP instances of size n. We saw in Assignment 1 that $f(n) = (n!)^{2n}$. Which of the following bounds are valid for $f(n)$? Choose all that apply.

   ○ $f(n) = O(n)$ ○ $f(n) = O((n!)^n)$ ○ $f(n) = O((n)^{3n})$
   ○ $f(n) = \Omega(n)$ ○ $f(n) = \Omega((n!)^n)$ ○ $f(n) = \Omega((n)^{3n})$

3. [2 points] Let $f(n)$ and $g(n)$ be any runtime functions that map non-negative integers to positive real numbers. Indicate whether the following statements must be true (for *all* such functions f and g), or could be false (for *some* such functions f and g).

   - If $f(n) = O(g(n))$ then $f(n) = o(n \cdot g(n))$.

     ○ True ○ False

   - If $\log f(n) = O(\log g(n))$ then $f(n) = O(g(n))$.

     ○ True ○ False

2 True/False++

For each of the following claims below, select whether the claim is True or False. If the claim is True, justify your answer by proving the claim is always True. If the claim is False, justify your answer by providing and briefly explaining a counterexample in which the claim is not True.

1. [5 points] **Claim**: All SMP instances have at most two stable matchings.

   Choose one: ○ **TRUE** ○ **FALSE**

   Justification (proof for True or counterexample for False):

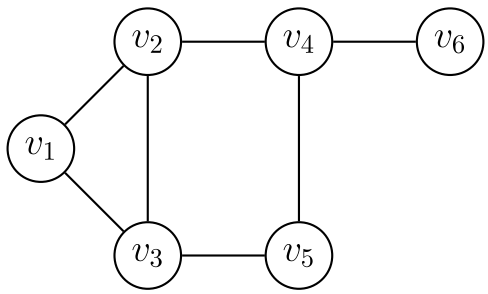
2. [5 points] Given: A connected graph G and a spanning tree T of G. **Claim**: The diameter of T is greater than or equal to the diameter of G.

   Choose one: ◯ **TRUE** ◯ **FALSE**

   Justification (proof for True or counterexample for False):

3 Reduction

Given a graph $G = (V, E)$, a **vertex cover** is a subset of vertices in V that includes at least one endpoint of every edge in the graph (note that the "endpoints" of an edge (u, v) are the vertices u and v). For example, the graph below has a vertex cover of size 3 given by $\{v_2, v_3, v_4\}$, but no vertex cover of size 2.



In this question, we consider the *Vertex Cover* problem (VCP):

> Given a graph $G = (V, E)$ and integer $k \geq 1$, does G have a vertex cover of size k? That is, is there a subset of k vertices in V with the property that every edge in E has an endpoint in the subset?

Recall also the **The Boolean satisfiability** (SAT) problem, defined as follows:

> The input is a collection of m clauses over r boolean variables X_1, X_2, $\ldots X_r$. Each clause is a disjunction of some of the variables or their complements.
>
> The problem consists of answering the question "Is there a way to assign truth values to each variable that makes **every** clause of the instance TRUE?

In the following questions you will complete a reduction from VCP to SAT.

1. [1 point] Our reduction defines nk variables $X_{i,j}$, for $1 \leq i \leq n$ and $1 \leq j \leq k$. What should the variable $X_{i,j}$ represent? Choose the **best** answer.

   ◯ Whether vertex v_i is in the potential vertex cover
   ◯ Whether vertex v_j is in the potential vertex cover
   ◯ Whether vertex v_i is the jth element of the potential vertex cover
   ◯ Whether vertex v_j is the ith element of the potential vertex cover

2. [2 points] Consider the following **partially complete** reduction from VCP to SAT. Define nk variables $X_{i,j}$, for $1 \leq i \leq n$ and $1 \leq j \leq k$.

   (A) For each a from 1 to k, add the clause:

   $$X_{1,a} \vee X_{2,a} \vee \ldots \vee X_{n,a}.$$

   (B) For every p from 1 to n and every distinct a, b from 1 to k, add the clause:

   $$\overline{X}_{p,a} \vee \overline{X}_{p,b}$$

   (C) For every distinct p, q from 1 to n and all a from 1 to k, add the clause:

   $$\overline{X}_{p,a} \vee \overline{X}_{q,a}$$

   For this reduction, Which of the following best describes the purpose of the clauses in (A)-(C)? Select all that apply.

   ◯ They ensure that the potential vertex cover contains exactly k vertices
   ◯ They ensure that every edge in E has at least one endpoint in the potential vertex cover
   ◯ They ensure that every vertex is in the potential vertex cover
   ◯ None of the above

3. [6 points] Complete the reduction by adding the necessary clauses in (D) below:

   (D) For each edge $e = (v_p, v_q)$, add the clause(s):

Appendix: Gale-Shapley Algorithm

```
set all s ∈ S and e ∈ E to free
while some free employer e hasn't made an offer to every student do
    s ← the highest-ranking student e hasn't made an offer to
    if s is free then
        hire(e, s)
    else
        e′ ← s's current employer
        if s prefers e to e′ then
            set e′ to free
            hire(e, s)
        endif
    endif
endwhile
return the set of pairs
```

Extra Space

If you write any part of your answers here, you must CLEARLY indicate it in the space allocated for that question.