CPSC 365 / ECON 365: Algorithms Yale University

Problem Set 1

Out: January 25, 2022 Due: February 1, 2022 at 2:30 pm EST

Instructions

Write your solution to the following problems carefully. Submit the PDF of your solution via Gradescope. Please start your solution to every question on a new page. Make sure to assign the correct page in your document corresponding to each problem. We recommend writing your solution in Latex. Handwritten solutions are accepted if they are clearly legible. Write your name and SID in your answer to Problem 0; do not write them anywhere else in your solution.

Collaboration policy: You may collaborate and discuss these problems with other students. However, you must first make an honest effort to solve these problems by yourself. You may discuss hints and try to solve problems together. But you may not explicitly provide the solution to a problem to other students; nor receive the solution from them. You must write your solution independently, and make sure you understand your solution. You must list all your collaborators (anyone with whom you discussed any part of these problems). You may consult the course textbook and other references related to the class. However, you are forbidden from searching for these problems on the Internet. You must list any resources you consult (beyond the course textbook). You must also follow the Yale academic integrity policy.

0 Your Information

On the first page of your submission, include the following information (but do not include these anywhere else in your solution). Also certify that you have followed the collaboration policy. Your solution to the remainder of the problems should start on a new page.

(a) Your name.

(b) Your SID.

(c) A list your collaborators and any outside resources you consulted for this problem set. If none, write "None".

(d) Certify that you have followed the academic integrity and collaboration policy as written above.

(e) How many hours did you spend in this problem set?

1 Logic

1.1 Propositional logic

In this problem, let P and Q be arbitrary logical propositions that may depend on some input x or y. For each of the following logical equivalence assertions, either: (i) prove it is true (prove both sides are always equal for all P and Q); or (ii) give a counterexample to show it is false (i.e., an example of P and Q such that one side of the equivalence is true and the other is false, and a short justification that it is indeed a counterexample).

(a) $\forall x \, \exists y \, P(x, y) \quad \equiv \quad \forall y \, \exists x \, P(x, y)$

(b) $\forall x \, (P(x) \vee Q(x)) \quad \equiv \quad \neg(\exists x \, (\neg P(x) \wedge \neg Q(x)))$

(c) $P \Rightarrow \neg Q \quad \equiv \quad \neg P \Rightarrow Q$

(d) $(P \Rightarrow Q) \wedge (\neg P \Rightarrow \neg Q) \ \equiv \ P \Leftrightarrow Q$

1.2 Proof by contradiction

Recall that a *rational number* is a number q that can be written as the ratio of two integers: $q = \frac{a}{b}$ for some $a, b \in \mathbb{Z}$. A number is *irrational* if it is not a rational number.

Prove by contradiction that the sum of a rational number and an irrational number is irrational.

2 Asymptotics

2.1 Definition

Let $f(n)$ and $g(n)$ be the running times of some algorithms with input $n \in \mathbb{N}$. Here we are concerned with large input as $n \to \infty$.

(a) Suppose that $f = \Omega(g)$. Write down the definition (i.e., the exact quantified statement) of what this means.

(b) Let $f(n) = 10 \, n^{1/2}$ and $g(n) = n^{2/3}$. Prove that $g = \Omega(f)$ (i.e., give a proof of the similar quantified statement from part (a)).

(c) Assume $f = \Theta(g)$. Is it true that $\log f = \Theta(\log g)$? Either prove it or give a counterexample.

2.2 Comparison

In each of the following, indicate whether $f = O(g)$, or $f = \Omega(g)$, or both, in which case $f = \Theta(g)$. For each one, give a short justification for your answer.

(a) $f(n) = 10\,n + \log n, \qquad g(n) = n + (\log n)^2$

(b) $f(n) = 10 \log n, \qquad g(n) = \log(n^2)$

(c) $f(n) = n^{1.01}, \qquad g(n) = n(\log n)^2$

(d) $f(n) = n^2/\log n, \qquad g(n) = n(\log n)^2$

(e) $f(n) = (\log n)^{\log n}, \qquad g(n) = n/\log n$

(f) $f(n) = n^{1/2}, \qquad g(n) = 3^{\log_2 n}$

(g) $f(n) = n2^n, \qquad g(n) = 3^n$

(h) $f(n) = n!, \qquad g(n) = 2^n$

(i) $f(n) = \sum_{i=1}^{n} i^k, \qquad g(n) = n^{k+1} \qquad$ (for any $k \in \mathbb{N}$)

(j) $f(n) = \log(n!), \qquad g(n) = n \log n$

3 Computation

3.1 Geometric series

Let $c > 0$ be a positive real number. Let $f(n) = \sum_{i=0}^{n} c^i = 1 + c + c^2 + \cdots + c^n$. Show that:

(a) $f(n) = \Theta(1)$ if $c < 1$.

(b) $f(n) = \Theta(n)$ if $c = 1$.

(c) $f(n) = \Theta(c^n)$ if $c > 1$.

3.2 Recursion

Consider the following function. You may assume n is a power of 2, so $n = 2^k$ for some $k \geq 0$.

```
function f(n):
    if n > 1
        print.line("still going")
        f(n/2)
        f(n/2)
```

(a) Let $L(n)$ be the number of lines that the function above prints with input $n = 2^k$. Write the recurrence that $L(n)$ satisfies. What is the base case?

(b) Write the values of $L(n)$ for the first few $n = 2^k$, and guess what the solution $L(n)$ is.

(c) Prove by induction that your answer is correct.

4 Graphs

4.1 Bipartite graphs

Prove that a graph G is bipartite if and only if it has no cycle of odd length.

Recall to prove an *if and only if* statement you have to prove both directions of the implications. So this problem has two parts:

(a) Prove that if G is bipartite, then G has no cycle of odd length.

(b) Prove that if G has no cycles of odd length, then G is bipartite.

4.2 Trees

Suppose G is a tree (a connected graph with no cycle). Let $n \geq 1$ be the number of vertices of G.

Prove by induction that the number of edges in G is $n - 1$.

4.3 Cycles

Let $G = (V, E)$ be a graph on $n = |V|$ vertices with $m = |E|$ edges.

Suppose that $m \geq n$. Prove that G must contain a cycle.

5 Stable Matching

5.1 Hard example for stable matching

Run the stable matching algorithm (propose and reject with the men proposing) on the following example with $n = 4$ men $(1, 2, 3, 4)$ and 4 women (A, B, C, D).

Men's preference list:

1	A	B	C	D
2	B	C	A	D
3	C	A	B	D
4	A	B	C	D

Women's preference list:

A	2	3	4	1
B	3	4	1	2
C	4	1	2	3
D	1	2	3	4

5.2 A tight bound on the number of proposals

In lecture we showed that the stable matching algorithm must terminate after at most n^2 proposals. Prove a sharper bound showing that the algorithm must terminate after at most $n(n-1)+1$ proposals. Conclude that the example in part 1 is a worst case instance for $n=4$. How many days does the algorithm take on this instance?

Hint: Show that there is at most one man who proposes to the last woman in his list.