

## Problem 1

(a)

1. True
2. False
3. True
4. True
5. True

(b)

**Proof.** Proving by induction, we have base cases for  $n = 1$ ,  $n = 2$ ,  $n = 3$ .

$$a_1 < 2^1 \rightarrow 1 < 2$$

$$a_2 < 2^2 \rightarrow 2 < 4$$

$$a_3 < 2^3 \rightarrow 3 < 8$$

then we can assume

$$a_{n-1} < 2^{n-1}$$

$$a_{n-2} < 2^{n-2}$$

$$a_{n-3} < 2^{n-3}$$

Now we have that

$$\begin{aligned} 2^n &= 2 * 2^{n-1} = 2^{n-1} + 2^{n-1} = \\ 2^{n-1} + 2 * 2^{n-2} &= 2^{n-1} + 2^{n-2} + 2^{n-2} = \\ 2^{n-1} + 2^{n-2} + 2 * 2^{n-3} &= 2^{n-1} + 2^{n-2} + 2^{n-3} + 2^{n-3} \end{aligned}$$

and now we can compare  $a$  and  $2^n$

$$a_n = a_{n-1} + a_{n-2} + a_{n-3}$$

$$2^n = 2^{n-1} + 2^{n-2} + 2^{n-3} + 2^{n-3}$$

and if we substitute the assumptions to the comparisons

$$a_n < 2^{n-1} + 2^{n-2} + 2^{n-3}$$

$$a_n < 2^n$$

□

(c)

To compute the  $|V|$  distances above, run the Breadth first search algorithm on the following

graph:  $G^R$ .

## Problem 2

### Problem: 4.7

**Given:**  $n$  distinct jobs ( $J_n$ ) with  $p_i$  required time on SC and  $f_i$  required time on PC

**Goal:** schedule (supercomputer order) with as small of a completion time as possible.

Possible Greedy Algorithms:

1. Earliest completion time on the PC
2. Shortest runtime on SC

We can create an example for where the second doesn't work, and the correct Greedy algorithm is the descending PC time.

Using the exchange argument, we can prove using proof by cases.

We define the job orders such that all jobs are ordered by decreasing finish times such that  $i < j \rightarrow f_i \geq f_j$ .

We define an inversion as an ordering of two jobs  $J_i, J_j$  where the finish times,  $f_i$  and  $f_j$  are such that  $f_i < f_j$ .

Our cases are

1. OPT has no inversions
2. We can decrease the number of inversions in OPT to make it more similar to greedy.

**Proof.** Consider an optimal solution OPT where OPT is the most similar optimal solution to greedy.

□

### Claim

Consider an optimal solution OPT which is most similar to the greedy solution. We claim that OPT equals the greedy solution.

**Proof.** Suppose to the contrary that OPT is not equal to the greedy solution. Then OPT contains an inversion, and thus contains a adjacent inversion too. Thus, OPT contains some  $i = j + 1$  where  $f_i < f_j$ .

Let  $P$  be time that preprocessing starts for job  $j$ . Then we can define the finish time of the two jobs as

$$F_j = P + p_j + f_j$$

$$F_i = P + p_j + p_i + f_i$$

If we swap the inversion the finishing times become

$$F'_i = P + p_i + f_i < F_i = P + p_j + p_i + f_i$$

$$F'_j = P + p_i + p_j + f_j < F_i = P + p_j + p_i + f_i$$

Since the finishing time of each of the swapped order is at least as good as  $F_i$ , we have that swapping the adjacent inversions does not make OPT worse, meaning that this is an optimal solution closer to our Greedy Solution, which contradicts the definition of OPT. Therefore, OPT must equal Greedy.

□

The algorithm runs in  $O(n \log n)$  time, as it sorts the jobs which is log linear.