HW 0 - CS 120

Evan Jolley

August 2022

1

1.1

This program recursively makes its way down the tree, running once for every node it encounters. Due to the fact it is called on every parent and its children, the number of times it runs will be n. That means it will run in time $O(n)$.

1.2

We assume this for the purpose of contradiction:

$$\phi(V*) > \frac{n}{2}$$

where $V*$ is the vertex that when removed results in the minimum possible size for the largest resulting tree, and ϕ is the function that removes the vertex. Now we must find the contradiction this assumption causes to prove our initial rule.

Once the selected vertex is removed, there are $n - 1$ vertices remaining.

$\geq \frac{n+1}{2}$ of these vertices are in the largest remaining tree as $\phi(V*) > \frac{n}{2}$

That means the one or two smaller trees have $\leq \frac{n-3}{2}$ vertices total as $\frac{n+1}{2} + \frac{n-3}{2} = n - 1$ total vertices.

Move from the initial vertex to the adjacent vertex belonging to the $\geq \frac{n+1}{2}$ tree. The new removed vertex belonged to the larger tree, meaning that the remaining singular tree or newly divided pair of trees will have $\geq \frac{n+1}{2} - 1 = \frac{n-1}{2}$ vertices.

The formerly removed vertex will now belong to the initially smaller group, as we moved toward the larger side. Thus, this group will now have $\leq \frac{n-3}{2} + 1 = \frac{n-1}{2}$ vertices.

This leaves us with two groups of vertices, one with $\leq \frac{n-1}{2}$ and the other with $\geq \frac{n-1}{2}$. This clearly violates $\phi(V*) > \frac{n}{2}$ as both groups could have $\frac{n-1}{2}$. Thus,

$$\phi(V*) \leq \frac{n}{2}$$

1.3

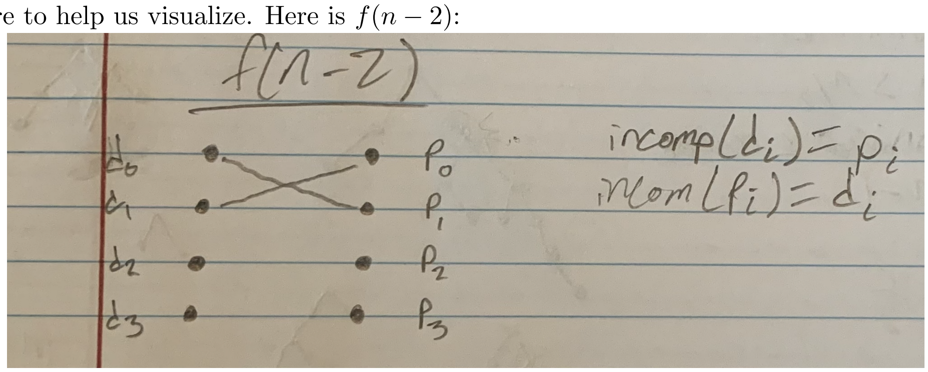
This program runs down the tree in a straight line, travelling from parent to child until it finds $V*$. It does not check every vertex, as it only follows a single path. The upper bound of the distance between the top and bottom of a tree is h by definition, so this program runs in time $O(h)$.

2

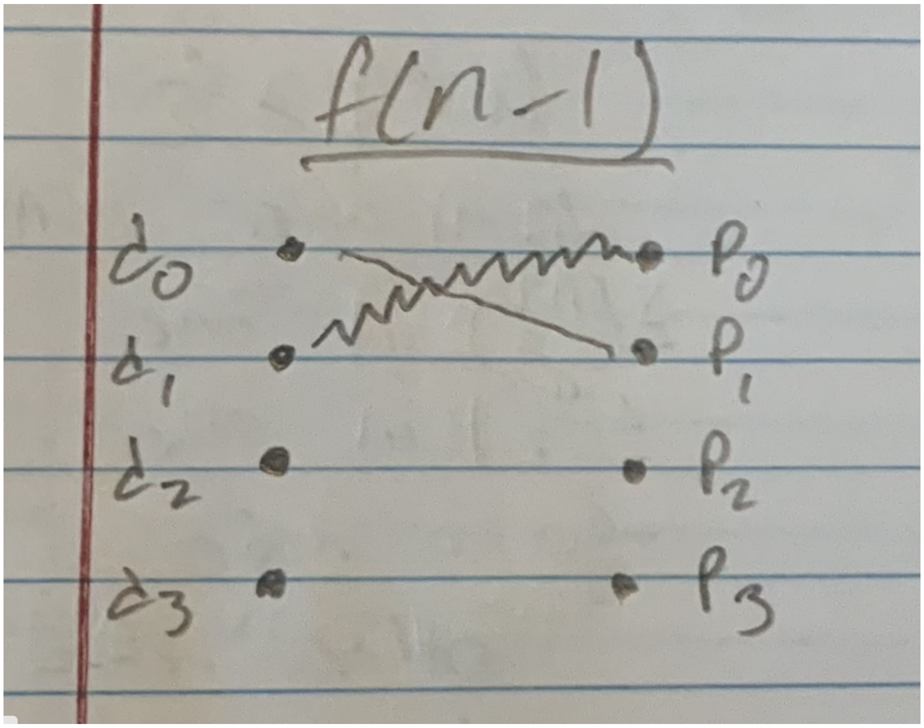
2.1

The $(n-1)$ term represents the number of patients that each donor can match with. Using the multiplication rule, we know that we can multiply this total to all of the number of different cases to find the total number of possibilities.

$f(n-2)$ and $f(n-1)$ can be seen as two different cases. Let's draw a picture to help us visualize. Here is $f(n-2)$:



This represents when two donors match with each others' incompatible patients. This results in the same problem with two less pairs, or, put more mathematically, $f(n-2)$, as illustrated by d2, d3, p2, and p3. The other case is $f(n-1)$:



This case represents all other possibilities: d1 matches with any patient other than p0. This sounds like $f(n)$ but with one less possibility. Thus, we can think of it as $f(n-1)$. d1 is either matched with p0 or it isn't, so this covers all possibilities. Also, note that the numbers and specific donors and patients used are arbitrary.

2.2

Basis: Let's prove a few cases first!

$$f(2) = 1$$

$$\frac{2!}{3} \leq 1 \leq \frac{2!}{2} \checkmark$$

$$f(3) = (3-1) * (1+0) = 2$$

$$\frac{3!}{3} \leq 2 \leq \frac{3!}{2} \checkmark$$

Induction hypothesis: Let's assume $\frac{k!}{3} \leq f(k) \leq \frac{k!}{2}$ for $2 \leq k \leq n$

Inductive step: Now, let's prove the assumption for $n+1$. We can solve the maximum and minimum cases separately.

Max:

$$f(n+1) = \frac{(n+1)!}{2}$$
$$(n) * (f(n) + f(n-1)) = f(n+1)$$
$$(n)(f(n) + f(n-1)) = \frac{(n+1)!}{2}$$
$$(n)(\frac{n!}{2} + \frac{(n-1)!}{2}) = \frac{(n+1)!}{2}$$
$$(n)(n! + (n-1)!) = (n+1)!$$
$$n * n! + n! = (n+1) * n!$$
$$n!(n+1) = (n+1)n!$$
$$n = n\checkmark$$

Min:

$$f(n+1) = \frac{(n+1)!}{3}$$
$$(n) * (f(n) + f(n-1)) = f(n+1)$$
$$(n)(f(n) + f(n-1)) = \frac{(n+1)!}{3}$$
$$(n)(\frac{n!}{3} + \frac{(n-1)!}{3}) = \frac{(n+1)!}{3}$$
$$(n)(n! + (n-1)!) = (n+1)!\checkmark$$

We proved the above statement in the maximum example!