Homework 2, CSCI 405

Your name here

April 22, 2024

Remember, you may work with your classmates but you must write up your own solutions and not copy each other. Show your work! **State your group members or that you worked alone.**

Show clear, concise solutions that are a combination of pseudocode and prose. Justify the correctness and analyze the running time of your algorithms.

Total points: 70

1. *[20 points]* Just like Cut-Rod and Fibonacci, we are going to analyze a recursive algorithm and use dynamic programming to make it more efficient. The algorithm evaluates the Legendre polynomial of degree n at a point x. (These polynomials are useful throughout physics and math; I have used them recently in my research on neural networks.) They satisfy the recurrence:

$$(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x), \tag{1}$$

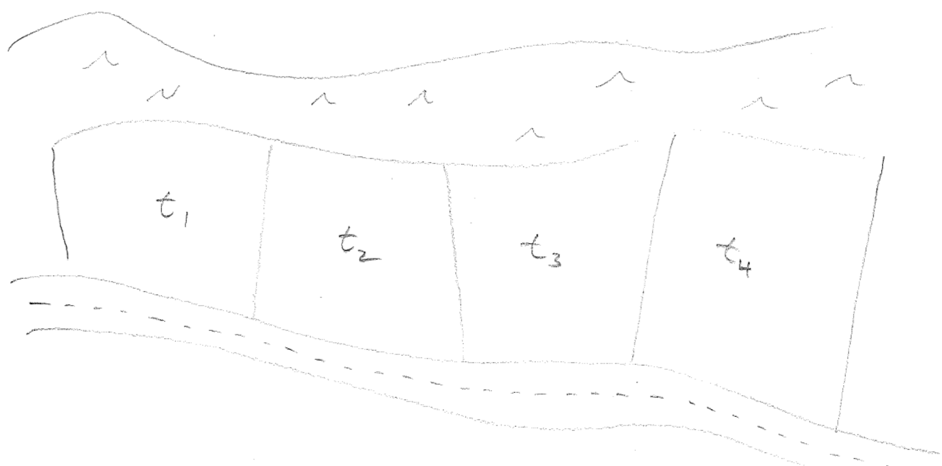
with base cases $P_0(x) = 1, P_1(x) = 2x - 1$. Each $P_n(x)$ is a degree n polynomial in x.

Here is a top-down algorithm for evaluation of $P_n(x)$:

```
Legendre(n, x):
if n == 0
    return 1
elif n == 1
    return 2 * x - 1
else
    return ((2 * n - 1) * x * Legendre(n - 1, x) - (n - 1) * Legendre(n - 2, x)) / n
```

1. *[5 points]* Draw the recursion tree for $n = 4$.
2. *[5 points]* Let $T(n)$ be the number of nodes for input n (the argument x isn't important). Show that $T(n) = \Omega((1.5)^n)$. Is this costly?
3. *[5 points]* Memo-ize the above program and analyze its performance.
4. *[5 points]* Give a bottom-up version and analyze its performance.

2. *[30 points]* You manage a team of truffle hunters in the Italian countryside along with their faithful hogs. The area you can forage is divided into discrete tracts $t_1, t_2, \ldots, t_n$ running from west to east. These are bordered to the north by a river and the south by the road (other areas are controlled by rival truffle hunters who don't take kindly to outsiders), so that t_i and t_{i+1} are adjacent for $i = 1$ to $n-1$. Here's an example for $n = 4$:



You have enough hogs that your limiting factor is that they must be kept apart by an empty tract, otherwise they become extremely aggressive and fight over the stinky fungi, and their handlers can only subdue them when unprovoked by another swine. Furthermore, some tracts are known to be more productive than others, so you assign a positive integer value $v_1, v_2, \ldots, v_n$ to each tract.

Goal: *Determine where to forage in order to maximize the value of the truffles you find while keeping the hogs from fighting.*

1. *[5 points]* Show that the following "most-valuable" greedy algorithm does not always find an optimal foraging plan:

   Start with S equal to the empty set
   While some tract is available
     Pick t_i with maximum v_i and add it to S
     Set t_i and its neighbors as unavailable
   Return S

2. *[5 points]* Show that the following is also not always an optimal strategy:

   Let S_1 be the set of tracts with i even
   Let S_2 be the set of tracts with i odd
   Return S_1 or S_2 depending on which has the greatest total value

3. *[20 points]* Give a dynamic programming algorithm that takes in a size-n truffle hunting problem (the values v_i are given and the layout of the tracts are the same, only the number changes) and returns an optimal foraging plan. It should run in polynomial-time, independent of the values. Points breakdown:

   - *[10 points]* Show optimal substructure for the truffle hunting problem.
   - *[8 points]* Provide a correct algorithm to solve the problem via dynamic programming.
   - *[2 points]* Analyze the runtime.

3. *[20 points]* There is a new videoconference and social network app named Snoom, wildly popular due to it's unique capability to enable chat while the user is asleep by tweaking the advertisements they see while "mindlessly" scrolling on their phone later to subconsciously imprint those messages in their brain. My mom and I use it so that I remember to always call.

Snoom, like most services these days, runs in the cloud which you can think of as a large collection of high-end but independent computers. However, their engineers have also invested in datacenter with very fast networking, i.e. a supercomputer, for certain costly computations. They periodically have to recalculate their social network's search index through a series of independent jobs $J_1, \ldots, J_n$. To complete job J_i, it must first be *preprocessed* on the datacenter which requires d_i time there and later c_i time *finishing* on one of the cloud machines.

There are many more than n cloud computers available, so every job can be finished in parallel, but the supercomputer can only work on one job at a time, serially. So when a job finishes preprocessing, it can proceed to the finishing stage regardless of whether there are other jobs finishing as well. Ensuring that all n jobs complete as quickly as possible is essential so that Snoom's social network grows as fast as possible (if you can't find your friends, how can you invite them to the network?), in line with the company's goal to maximize profit (ad revenue).

Your task:

1. *[8 points]* Devise an optimal polynomial-time scheduling algorithm that determines a schedule for the n jobs on the supercomputer so that the completion time of all the cloud machines is minimal. Hint: Think like the company's ruthless private equity backers.
2. *[2 points]* Analyze the runtime of your algorithm.
3. *[10 points]* Show that your algorithm returns the optimal solution. Try the "cut-and-paste" method.