CSCI 270 Lecture 7: Coin Changing

There are n denominations of coins $1 = d_1 < d_2 < ... < d_n$ (you have an unlimited amount of each coin). You have a target sum T. Determine the fewest number of coins needed to make change.

- Suppose I'm using American currency ($d_1 = 1, d_2 = 5, d_3 = 10, d_4 = 25$). What would be an optimal algorithm?
- Suppose my currency is $d_1 = 1, d_2 = 10, d_3 = 15$. Does the above algorithm still work?
- I need to break this up into bite-size decisions. What should my first decision be?
- What information do I need to pass down to the next level of recursion?

$CC[x]$ will be the fewest number of coins needed to make change for x cents.

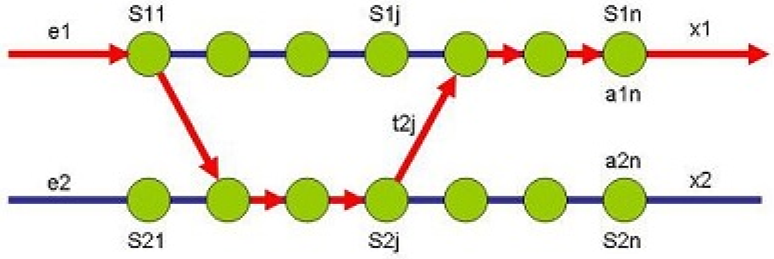
- If I select denomination d_i next, what recursive call should I make?
- If I select denomination d_i next, how does this change the number of coins I've used?
- At what point should I stop recursing?

$CC[x] = 1 + \min_i(CC[x - d_i])$
$CC[0] = 0$
$CC[x] = \infty$, for all $x < 0$.

- What order should I fill the array?
- Where is the answer stored?
- What is the runtime of the algorithm?
- Is this a polynomial-running time algorithm?

Assembly-Line Scheduling

There are 2 assembly lines, each with n stations. The ith station on each line is denoted $S_{1,i}$ and $S_{2,i}$. An automobile starts at your choice of $S_{1,1}$ or $S_{2,1}$. Station $S_{i,j}$ has a processing time $a_{i,j}$. After station $S_{i,j}$, the automobile can stay on the same line and go to $S_{i,j+1}$ at no cost, or switch lines at cost $t_{i,j}$.



- If we want to find the shortest path across the assembly lines, what basic algorithm could we use to solve this?
- We need to break this up into a series of ordered decisions. What should our first decision be? Our second decision?
- At each level of the recursion, we will be answered a single decision. What information changes at each level of the recursion, and thus must be passed down as parameters?
- If I am at $S_{i,j}$, what cost will I incur regardless of my choice?
- If I am at $S_{i,j}$, and I decide to stay on the same line, what station will I go to next?
- If I am at $S_{i,j}$, what cost will I incur if I decide to switch lines?

Define: ALS$[i,j]$ will store the length of the shortest path from $S_{i,j}$ to the exit.

$\text{ALS}[1,j] = a_{i,j} + \min(\text{ALS}[1,j+1], t_{1,j} + \text{ALS}[2,j+1])$

$\text{ALS}[2,j] = a_{i,j} + \min(\text{ALS}[2,j+1], t_{2,j} + \text{ALS}[1,j+1])$

- What is the base case?
- What order do I fill in the array?
- Where is the answer stored in the completed array?
- What is the runtime of this algorithm?
- How do I determine the actual path through the assembly lines?