CSCI 433 Homework 8 Solutions

Exercises 8.1.

1. Both techniques solve a problem by dividing it into several subproblems.

In divide-and-conquer, smaller subproblems do not overlap. Their solutions are not stored for reuse.

In dynamic programming, smaller subproblems overlap. Their solutions are stored for reuse.

6. Let $F(n)$ be the maximum price for a given rod of length n.

We have the following recurrence for its values:

$$F(n) = \max_{1 \le j \le n} \{ p_j + F(n-j) \}, \quad n > 0$$

$$F(0) = 0.$$

We can fill a 1D array with $n+1$ consecutive values of F.

Time efficiency $\Theta(n^2)$

Space efficiency $\Theta(n)$.

12 (H).

a. $P(i,j)$: the probability of A winning the series if A needs i more games to win the series and B needs j more games to win the series

If A wins the game, which happens with probability p, A needs $i-1$ more wins to win the series. B still needs j wins.

If A looses the game, which happens with probability $1-p$, A will still need i more wins and B needs $j-1$ wins.

$\Rightarrow$ Recurrence

$$P(i,j) = pP(i-1,j) + (1-p)P(i,j-1), \quad i,j>0$$

$$P(0,j) = 1, \quad j>0$$

$$P(i,0) = 0, \quad i>0$$

b

$p = 0.4$

i \ j	0	1	2	3	4
0		1	1	1	1
1	0	0.4	0.64	0.784	0.8704
2	0	0.16	0.352	0.5248	0.66304
3	0	0.064	0.1792	0.31744	0.45568
4	0	0.0256	0.08704	0.1792	0.289792

$P[4,4) \approx 0.29$

c.

```
Algorithm WorldSeries(n, p)
  // n: # of wins needed
  // p: team A wins a particular game.

  q ← 1-p
  for j ← 1 to n do
      P[0, j] ← 1
  for i ← 1 to n do
      P[i, 0] ← 0
      for j ← 1 to n do
          P[i, j] = p * P[i-1, j] + q * P[i, j-1]
  return P[n, n]
```

Time efficiency: $\Theta(n^2)$, Space efficiency $\Theta(n^2)$

Exercises 8.2

2. a. Algorithm DPKnapsack (w[1..n], v[1..n], W)

```
// Input:  w[1..n] — weights
           v[1..n] — values of n items
           W — knapsack capacity
// Output: F[0..n, 0..W]
           F[n,W] — values of an optimal subset

for i ← 0 to n do F[i,0] ← 0
for j ← 1 to W do F[0,j] ← 0
for i ← 1 to n do
    for j ← 1 to W do
        if j-w[j] ≥ 0
            F[i,j] ← max{ F[i-1,j], v[i] + F[i-1, j-w[i]] }
        else F[i,j] ← F[i-1,j]
return F[0..n, 0..W]
```

b. Algorithm Optimal Knapsack ($w[1..n]$, $v[1..n]$, $F[0..n, 0..W]$)

// Input: $w[1..n]$, $v[1..n]$, $F[0..n, 0..W]$ generated by DPKnapsack.

// Output: $L[1..k]$ of the items composing an optimal solution

```
k ← 0
j ← W
for i ← n down to 1 do
    if F[i,j] > F[i-1, j]
        k ← k+1;  L[k] ← i   // include i
        j ← j - w[i]
return L.
```

Exercises 8.3.

4. Precompute $S_k = \sum_{s=1}^{k} P_s$, $k = 1, 2, \cdots, n$.

$S_0 = 0$.

Then $\sum_{s=i}^{j} P_s = S_j - S_{i-1}$.

5. False.

Find a counter example.

7 a. $b(n)$: # of binary trees with n nodes.

If the left subtree of a binary tree with n nodes has k nodes,

the right subtree must have $n-1-k$ nodes.

$$b(n) = \sum_{k=0}^{n-1} b(k)\, b(n-1-k), \quad n > 0$$

$$b(0) = 1$$

b.

n	0	1	2	3	4	5
$b(n)=c(n)$	1	1	2	5	14	42

c. $b(n) = c(n) = \binom{2n}{n}\frac{1}{n+1} = \frac{(2n)!}{(n!)^2}\frac{1}{n+1}$

$$\approx \frac{\sqrt{2\pi 2n}\left(\frac{2n}{e}\right)^{2n}}{\left[\sqrt{2\pi n}\left(\frac{n}{e}\right)^{n}\right]^2}\frac{1}{n+1} = \frac{\sqrt{4\pi n}\left(\frac{2n}{e}\right)^{2n}}{2\pi n\left(\frac{n}{e}\right)^{2n}}\frac{1}{n+1}$$

$$= \frac{1}{\sqrt{\pi n}}\left(\frac{2n/e}{n/e}\right)^{2n}\frac{1}{n+1} = \frac{1}{\sqrt{\pi n}}4^n\frac{1}{n+1} \in \Theta(4^n n^{-3/2})$$

$\Rightarrow$ Brute force is infeasible.

CSCI 433 Homework 8 Solutions

Exercises 8.4

1 $$R^{(0)} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad R^{(1)} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$R^{(2)} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad R^{(3)} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$R^{(4)} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

7. $$D^{(0)} = \begin{bmatrix} 0 & 2 & \infty & 1 & 8 \\ 6 & 0 & 3 & 2 & \infty \\ \infty & \infty & 0 & 4 & \infty \\ \infty & \infty & 2 & 0 & 3 \\ 3 & \infty & \infty & \infty & 0 \end{bmatrix}, \quad D^{(1)} = \begin{bmatrix} 0 & 2 & \infty & 1 & 8 \\ 6 & 0 & 3 & 2 & 14 \\ \infty & \infty & 0 & 4 & \infty \\ \infty & \infty & 2 & 0 & 3 \\ 3 & 5 & \infty & 4 & 0 \end{bmatrix}$$

$$D^{(2)} = \begin{bmatrix} 0 & 2 & 5 & 1 & 8 \\ 6 & 0 & 3 & 2 & 14 \\ \infty & \infty & 0 & 4 & \infty \\ \infty & \infty & 2 & 0 & 3 \\ 3 & 5 & 8 & 4 & 0 \end{bmatrix}, \quad D^{(3)} = \begin{bmatrix} 0 & 2 & 5 & 1 & 8 \\ 6 & 0 & 3 & 2 & 14 \\ \infty & \infty & 0 & 4 & \infty \\ \infty & \infty & 2 & 0 & 3 \\ 3 & 5 & 8 & 4 & 0 \end{bmatrix}$$

$$D^{(4)} = \begin{bmatrix} 0 & 2 & 3 & 1 & 4 \\ 6 & 0 & 3 & 2 & 5 \\ \infty & \infty & 0 & 4 & 7 \\ \infty & \infty & 2 & 0 & 3 \\ 3 & 5 & 6 & 4 & 0 \end{bmatrix}, \quad D^{(5)} = \begin{bmatrix} 0 & 2 & 3 & 1 & 4 \\ 6 & 0 & 3 & 2 & 5 \\ 10 & 12 & 0 & 4 & 7 \\ 6 & 8 & 2 & 0 & 3 \\ 3 & 5 & 6 & 4 & 0 \end{bmatrix} = D.$$

11. For each pair of the straws, determine whether the straws intersect.

Record the information in a boolean $n \times n$ matrix.

Find the transitive closure of this matrix using DFS or BFS.

Time efficiency: $\Theta(n^2)$.