CISC 610-90- O-2018/Late Fall
Assignment 4

b) Analysis1 (D&C)

Time to solve the 4 subprograms of size $n/4$ is $4T(n/4)$. As the combination is in $O(n^2)$ time, we get the recurrence:

$$T(n) = 4T(n/4) + O(n^2).$$

So the recurrence is of format for Master theorem for divide and conquer. From recurrence we have $a = 4$, $b = 4$, $k = 2$, $i = 0$, and $c = 2$.

Since $c = 2 > log_b a = log_4 4 = 1$, we get the solution:

$$T(n) = O(n^c) = O(n^2).$$

c) Analysis2 (D&C)

Time to solve the one sub-program of size $n - 2$ is $1T(n - 2)$. As the combination is in $O(n^2)$ time, we get the recurrence:

$$T(n) = 1T(n - 2) + O(n^2).$$

So the recurrence is of format for Master theorem for subtract and conquer. From recurrence we have $a = 1$, $b = 2$, and $k = 2$.

Since $a = 1 > 0$, $b = 2 > 0$, $k = 2 \geq 0$ and $a = 1$, we get the solution:

$$T(n) = O(n^{k+1}) = O(n^3).$$

d) Floyd (Dynamic prog)

Follow the Floyd Algorithm to calculate array D which contains the lengths of the shortest paths, and array P which contains the highest indexes of intermediate vertexes on the shortest paths.:

- $k = 0$:

$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 0 & 1 & \infty & 1 & 5 \\ 2 & 9 & 0 & 3 & 2 & \infty \\ 3 & \infty & \infty & 0 & 4 & \infty \\ 4 & \infty & \infty & 2 & 0 & 3 \\ 5 & 3 & \infty & \infty & \infty & 0 \end{array} \qquad \begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 & 0 & 0 \\ 5 & 0 & 0 & 0 & 0 & 0 \end{array}$$

D $\qquad$ P

- $k = 1$:

$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 0 & 1 & \infty & 1 & 5 \\ 2 & 9 & 0 & 3 & 2 & 14 \\ 3 & \infty & \infty & 0 & 4 & \infty \\ 4 & \infty & \infty & 2 & 0 & 3 \\ 5 & 3 & 4 & \infty & 4 & 0 \end{array} \qquad \begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 1 \\ 3 & 0 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 & 0 & 0 \\ 5 & 0 & 1 & 0 & 1 & 0 \end{array}$$

D $\qquad$ P

- $k = 2$:

$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 0 & 1 & 4 & 1 & 5 \\ 2 & 9 & 0 & 3 & 2 & 14 \\ 3 & \infty & \infty & 0 & 4 & \infty \\ 4 & \infty & \infty & 2 & 0 & 3 \\ 5 & 3 & 4 & 7 & 4 & 0 \end{array} \qquad \begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 0 & 0 & 2 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 1 \\ 3 & 0 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 & 0 & 0 \\ 5 & 0 & 1 & 2 & 1 & 0 \end{array}$$

D $\qquad$ P

- $k = 3$:

$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 0 & 1 & 4 & 1 & 5 \\ 2 & 9 & 0 & 3 & 2 & 14 \\ 3 & \infty & \infty & 0 & 4 & \infty \\ 4 & \infty & \infty & 2 & 0 & 3 \\ 5 & 3 & 4 & 7 & 4 & 0 \end{array} \qquad \begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 0 & 0 & 2 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 1 \\ 3 & 0 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 & 0 & 0 \\ 5 & 0 & 1 & 2 & 1 & 0 \end{array}$$

D $\qquad$ P

- $k = 4$:

$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 0 & 1 & 3 & 1 & 4 \\ 2 & 9 & 0 & 3 & 2 & 5 \\ 3 & \infty & \infty & 0 & 4 & 7 \\ 4 & \infty & \infty & 2 & 0 & 3 \\ 5 & 3 & 4 & 6 & 4 & 0 \end{array} \qquad \begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 0 & 0 & 4 & 0 & 4 \\ 2 & 0 & 0 & 0 & 0 & 4 \\ 3 & 0 & 0 & 0 & 0 & 4 \\ 4 & 0 & 0 & 0 & 0 & 0 \\ 5 & 0 & 1 & 4 & 1 & 0 \end{array}$$

D P

- $k = 5$:

$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 0 & 1 & 3 & 1 & 4 \\ 2 & \mathbf{8} & 0 & 3 & 2 & 5 \\ 3 & 10 & 11 & 0 & 4 & 7 \\ 4 & 6 & 7 & 2 & 0 & 3 \\ 5 & 3 & 4 & 6 & 4 & 0 \end{array} \qquad \begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 0 & 0 & 4 & 0 & 4 \\ 2 & 5 & 0 & 0 & 0 & 4 \\ 3 & 5 & 5 & 0 & 0 & 4 \\ 4 & 5 & 5 & 0 & 0 & 0 \\ 5 & 0 & 1 & 4 & 1 & 0 \end{array}$$

D P

The shortest distance from v_2 to v_1 is 8 shown as shown in array D.

To find the path, apply print path algorithm in the solution array P. For the path function in Algorithm 3.5, we have

$P[2][1] = 5 \neq 0$
$\Rightarrow$ recursively call path$(2, 5)$
    $P[2][5] = 4 \neq 0$
    $\Rightarrow$ recursively call path$(2, 4)$
        $P[2][4] = 0$ END
    print v_4
print v_5

Therefore, the shortest path from v_2 to v_1 is:

$$v_2 \rightarrow v_4 \rightarrow v_5 \rightarrow v_1$$

e) OptimumBinarySearchTree

The average search times for the tress in figure are:

$$1p_1 + 2p_2 + 3p_3 + 4p_4 = 0.2 + 0.2 + 1.2 + 1.2 = 2.8 \tag{1}$$

$$2p_1 + 1p_2 + 2p_3 + 3p_4 = 0.4 + 0.1 + 0.8 + 0.9 = 2.2 \tag{2}$$

$$2p_1 + 2p_2 + 1p_3 + 3p_4 = 0.4 + 0.2 + 0.4 + 0.9 = 1.9 \tag{3}$$

The third tree is optimal.