CS5200 Homework 2 Dynamic Programming

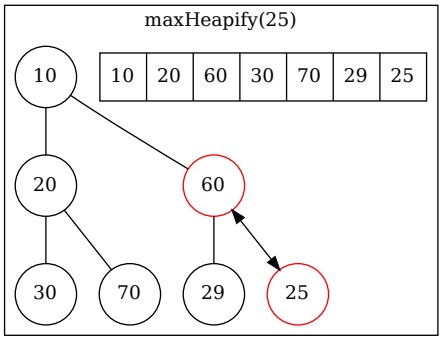
Adam McNeil

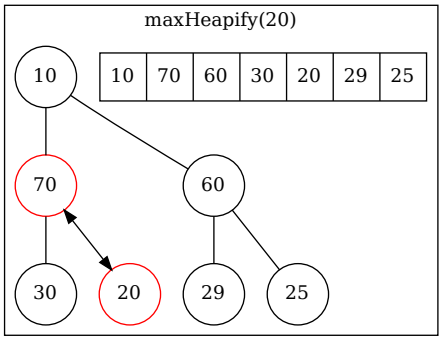
Question 1)

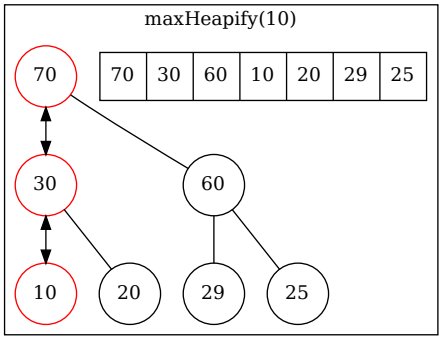
max heapify

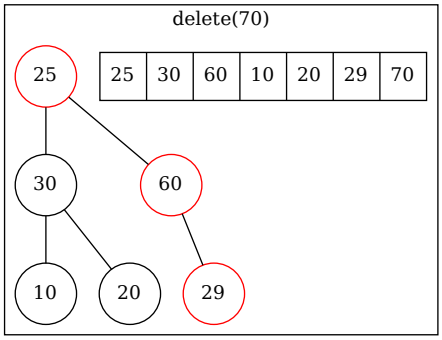
Call max heapify on all the internal nodes starting at the bottom

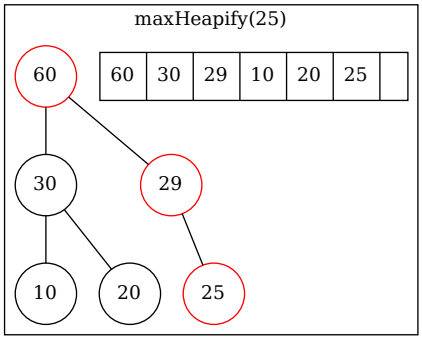
maxHeapify(25)

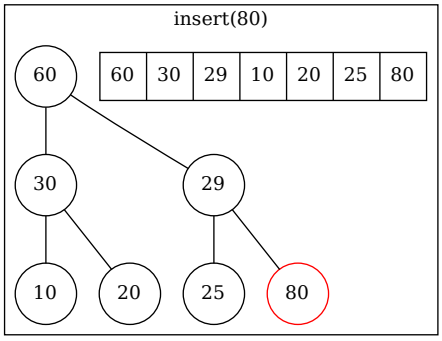


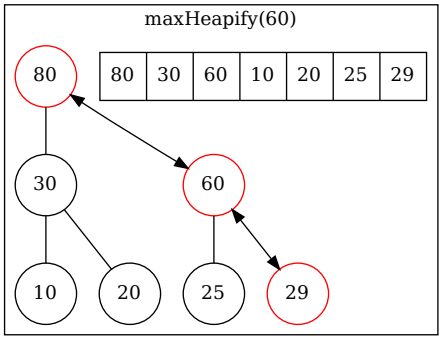


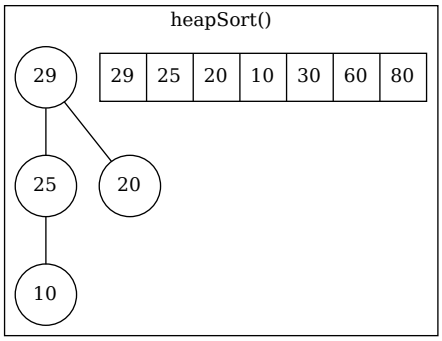

maxHeapify(10)
70
70 30 60 10 20 29 25
30
60
10
20
29
25

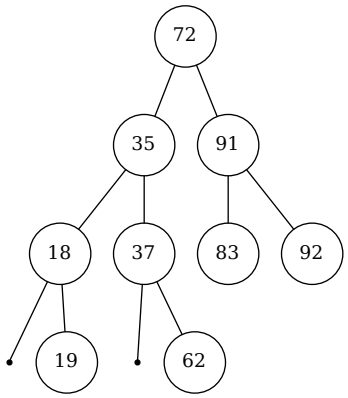

delete(70)
25
25 30 60 10 20 29 70
30
60
10
20
29


maxHeapify(25)
60
60 30 29 10 20 25
30
29
10
20
25

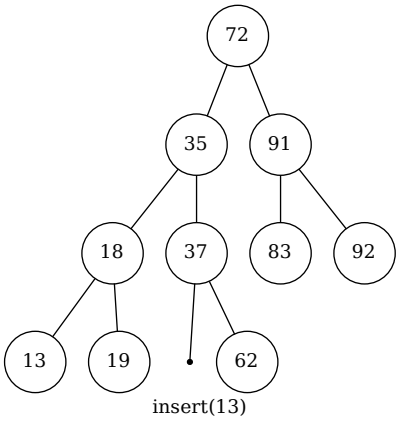

insert(80)
60
60 30 29 10 20 25 80
30
29
10
20
25
80

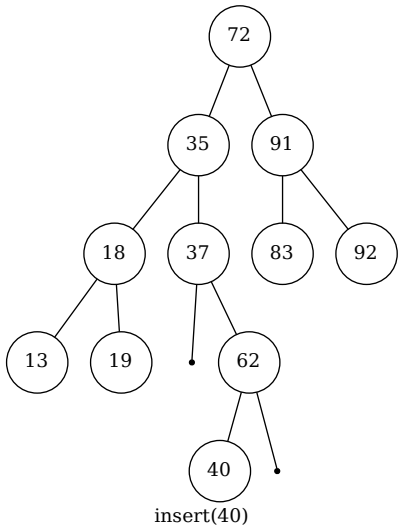

maxHeapify(60)
80
80 30 60 10 20 25 29
30
60
10
20
25
29


heapSort()
29
29 25 20 10 30 60 80
25
20
10

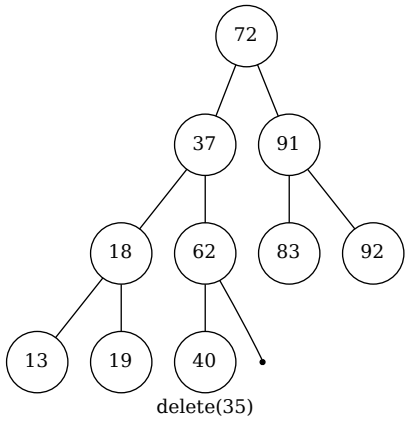


Question 2)
Pre-order: 72 35 18 19 37 62 91 83 92
In-order: 18 19 35 37 62 72 83 91 92
Post-order: 19 18 62 37 35 83 92 91 72





insert(40)



delete(35)

Question 3)
$p_0 = 4 \;\; p_1 = 10 \;\; p_2 = 3 \;\; p_3 = 12 \;\; p_4 = 7$

	1	2	3	4		1	2	3	4
4	0	0	252	0	4			3	0
3	0	360	0		3		2	0	
2	120	0			2	1	0		
1	0				1	0			

m(1, 3) i=1 j=3

k=1
m(1, 1) + m(2, 3) + p_0 p_1 p_3
0 + 360 + 4*10*12 = 840

k=2
m(1, 2) + m(3, 3) + p_0 p_2 p_3
120 + 0 + 4*3*12 = 264

m(2, 4) i=2 j=4
k=2
m(2, 2) + m(3, 4) + p_1 p_2 p_4
0 + 252 + 10*3*7 = 462

k=3
m(2, 3) + m(4, 4) + p_1 p_3 p_4
120 + 0 + 10*12*7 = 462

	1	2	3	4
4	0	462	252	0
3	264	360	0	
2	120	0		
1	0			

	1	2	3	4
4		2	3	0
3	2	2	0	
2	1	0		
1	0			

m(1, 4) i=1 j=4
k=1
m(1, 1) + m(2, 4) + p_0 p_1 p_4
0 + 462 + 4*10*7 = 742

k=2
m(1, 2) + m(3, 4) + p_0 p_2 p_4
120 + 252 + 4*3*7 = 456

k=3
m(1, 3) + m(4, 4) + p_1 p_3 p_4
264 + 0 + 4*12*7 = 600

	1	2	3	4
4	456	462	252	0
3	264	360	0	
2	120	0		
1	0			

	1	2	3	4
4	2	2	3	0
3	2	2	0	
2	1	0		
1	0			

$(A_1\ A_2)\ (A_3\ A_4)$

Question 4)

			CACMYCCA								
		_	C	A	C	M	Y	C	C	A	
	_	↖0	←0	←0	←0	←0	←0	←0	←0	←0	
MCMAMYCCMAY	M	↑0	←0	←0	←0	↖1	←1	←1	←1	←1	
	C	↑0	↖1	←1	←1	←1	←1	↖2	←2	←2	
	M	↑0	↑1	←1	←1	↖2	←2	←2	←2	←2	
	A	↑0	↑1	↖2	←2	←2	←2	←2	←2	↖3	
	M	↑0	↑1	↑2	←2	↖3	←3	←3	←3	←3	
	Y	↑0	↑1	↑2	←2	↑3	↖4	←4	←4	←4	
	C	↑0	↑1	↑2	↑3	←3	↑4	↖5	←5	←5	
	C	↑0	↑1	↑2	↑3	←3	↑4	↑5	↖6	←6	
	M	↑0	↑1	↑2	↑3	↖4	←4	↑5	↑6	←6	
	A	↑0	↑1	↑2	↑3	↑4	←4	↑5	↑6	↖7	
	Y	↑0	↑1	↑2	↑3	↑4	↖5	↑5	↑6	↑7	CAMYCCA

			CACMYCCA								
		_	C	A	C	M	Y	C	C	A	
	_	0	0	0	0	0	0	0	0	0	
MCMAMYCCMAY	M	0	0	0	0	1	0	0	0	0	
	C	0	1	0	1	0	0	1	1	0	
	M	0	0	0	0	2	0	0	0	0	
	A	0	0	1	0	0	0	0	0	1	
	M	0	0	0	0	1	0	0	0	0	
	Y	0	0	0	0	0	2	0	0	0	
	C	0	1	0	1	0	0	3	1	0	
	C	0	1	0	1	0	0	1	4	0	
	M	0	0	0	0	2	0	0	0	0	
	A	0	0	1	0	0	0	0	0	1	
	Y	0	0	0	0	0	1	0	0	0	MYCC

Question 5)

i	0	1	2	3	4
p_i	0	0.05	0.12	0.3	0.2
q_i	0.07	0.07	0.07	0.06	0.06

$$w[i, j] = w[i, j\text{-}1] + p_j + q_j$$

$w[1, 1] = w[1, 0] + p_1 + q_1 = 0.07 + 0.05 + 0.07 = 0.19$
$w[3, 2] = w[3, 1] + p_2 + q_2 = 0.26 + 0.30 + 0.06 = 0.62$
$w[4, 4] = w[4, 3] + p_4 + q_4 = 0.06 + 0.20 + 0.06 = 0.32$

w	1	2	3	4	5
4	1.00	0.88	0.69	0.32	0.06
3	0.74	0.62	0.43	0.06	
2	0.52	0.26	0.07		
1	0.19	0.07			
0	0.07				

r is from i to j
save the lowest r to root table and record the lowest value in the c table
c[i, j] = c[i, r-1] + c[r+1, j] + w[i, j]

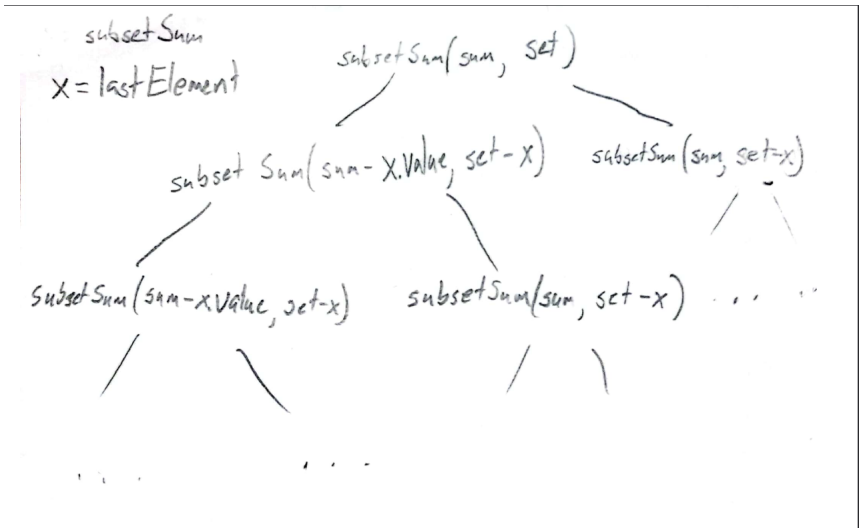
r = 1
c[1, 1] = c[1, 0] + c[2, 1] + w[1, 1] = 0.07 + 0.07 + 0.19 = 0.33

r = 1
c[1, 2] = c[1, 0] + c[2, 2] + w[1, 2] = 0.07 + 0.40 + 0.52 = 0.99
r = 2
c[1, 2] = c[1, 1] + c[3, 2] + w[1, 2] = 0.33 + 0.07 + 0.52 = 0.92

c	1	2	3	4	5
4	2.36	1.76	1.20	0.44	0.06
3	1.63	1.08	0.56	0.06	
2	0.92	0.40	0.07		
1	0.33	0.07			
0	0.07				

root	1	2	3	4
4	3	3	3	4
3	2	3	3	
2	2	2		
1	1			

Bonus:
optimal substructure the solution either contains the number or does not contain the number



Pseudo code:

```
sumOfSubset(set, n)
    listOfSums = [0]
    for i in set:
        for j in listOfSums:
            if i + j is not in listOfSums:
                listOfSums.addFront(i+j)
    return is n in listOfSums
```