# Mir Hossain

# CS477

# HW7

Mr Hassan
CS477
H/W#7

1a) Customer = i
time job = $t_i$
weight = $w_i$
Goal: Minimize $\sum_{i=1}^{n} w_i C_i$

Observation: Go for the greedy approach, the greatest weight/time should be finished first

Algorithm: $\frac{w_1}{t_1} > \frac{w_2}{t_2} > \frac{w_3}{t_3} \cdots \frac{w_n}{t_n}$, most important
Print jobs are completed first, so we would have the most in customers possible

Proof: $\sum_{i=1}^{n} w_i C_i = w_1 C_1 + w_2 C_2 + w_3 C_3 \cdots w_n C_n$
↳ weight to Processing time ratio.
The algorithm has to take properties from a PriorityQueue, so the most important jobs are completed first

Greedy Solution:
$T_1$: First Page Prints in 1 second
$T_2$: Second Page prints in 2 seconds

$Job_A = 10$ (weight)
$Job_B = 2$ (weight)

$A_{greedy} = (Job_A \cdot T_1) + (Job_B \cdot T_2) = (10 \cdot 1) + (2 \cdot 2) = 10 + 4 = 14$
$A_{other} = (Job_B \cdot T_1) + (Job_A \cdot T_2) = (2 \cdot 1) + (10 \cdot 2) = 2 + 20 = 22$ (1)

Next Page =>

$A_{greedy} = 14$; $A_{other} = 22$

$A_{greedy} < A_{other}$

Conclusion: The greedy solution ($A_{greedy}$) will always be the most efficient solution. If the printer were to print a page every second, then time would be following a linear growth. By sorting the print jobs as a priority queue, where the most important jons are printed first, then we will have the smallest weighted sum possible.

2a) - Workers = n
- one shift every week
- Shifts are contgious [Start, Final]
- Multiple Shifts possible
* Goal: Create an Advisory Comitee that can observe workers. Most amount of workers w/ least amount of Advisory Workers

key: C = Advisory Comitee
W = workers
P = set of workers w/ overlap

Observation: Go for Greedy Solution by Utilizing Interval Partionly maximize number of workers while minimizing number of Advisors

Algorithm:
1) W = workers
2) Sort(Shifts) // earliest finishy Shift → Latest finishy Shift
3) for (i > Shifts) // i in set W = {worker$_1$, worker$_2$, worker$_3$}
4) P = {w ∈ W | overlaps in interval i} // P is the set
5) p ∈ P // latest finish time, and add that to set C
(C = Comittee)
6) Delete interval i in set W that has overlap with p ∈ P
7) Loop(w ∈ W) // Loop till W is empty

Conclusion:

Let $C = \{C_1, C_2, C_3, C_4 \ldots C_j\}$, where C represents advisors ordered from latest earliest shift (Mid Shift) to latest shift. The optimal solution must be $k < j$ advisors. Using proof by contradiction, we try to prove the greedy solution is not the optimal solution.

Let $O = \{O_1, O_2, O_3, \ldots O_k\}$, where O represents the "other" solution where we order the earliest finish time to latest finish time. Assume $\{C_n\}$ has a later or equal finish time than $\{O_n\}$:

$\{C_1, C_2, C_3, C_4 \ldots C_n\} \geq \{O_1, O_2, O_3, O_4 \ldots O_n\}$ For every $i < k$, the $(C_i)$ advises $(O_i)$ & $C_k$ advises $(O_k)$.

Since $C = \{C_1, C_2, C_3 \ldots C_k\}$, then this implies that $j = k$. However the optimal solution is $k < j$. Because of proof by contradiction, greedy solution is optimal solution.

3.) I=75, U=200, B=25, S=275, C=50, H=100, M=25, P=25

$B \le m < C < I < H < U < P < S$

C = Alphabet of Chars

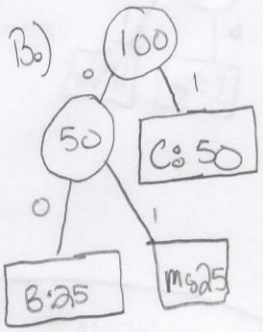
f(c) = frequency of c

$d_t(c)$ = depth of c's leaf in tree

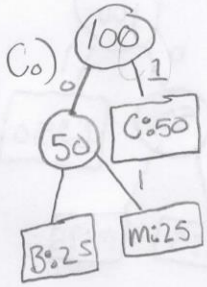
$B(T) = \sum f(c)\, d_t(c)$

A.)


B.)

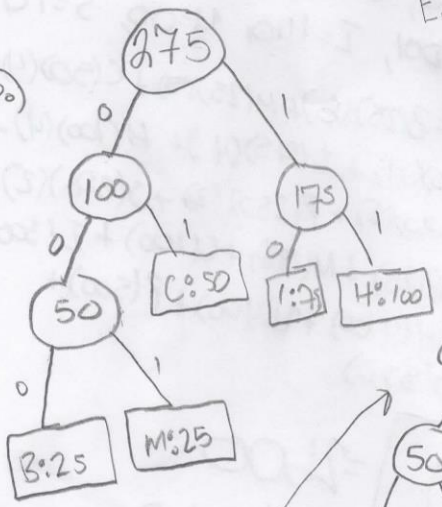



C.)

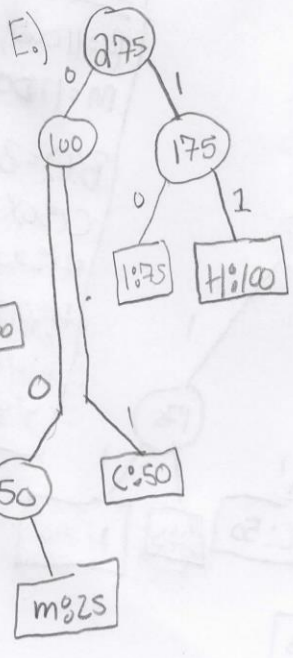


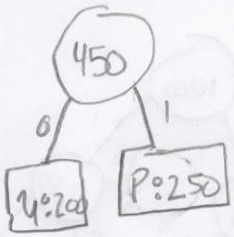


D.)

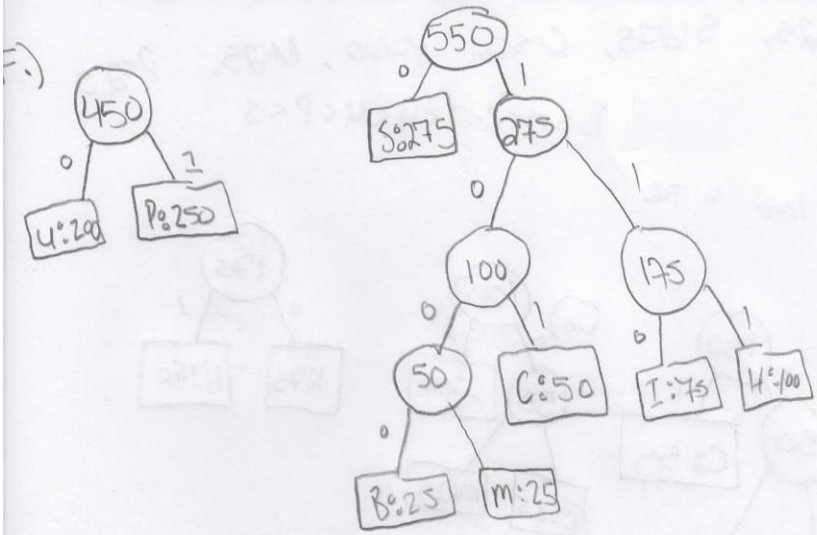



E.)

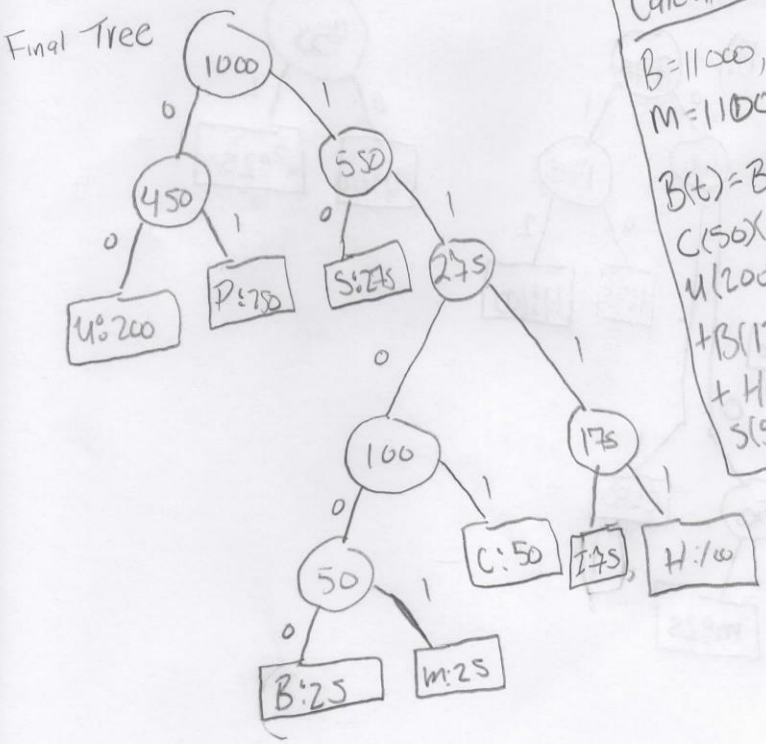



Didn't think about spacing




Final Tree




Calculation:

B=1100, C=1101, H=1111, P=01,
M=11001, I=1110, U=00, S=10

B(t)=B(25)(B)+M(25)(5)+C(50)(4)+
C(50)(4)+I(75)(4)+H(100)(4)+
U(200)(2)+P(250)(2)+S(275)(2)
+B(125)+M(125)+C(200)+I(300
+H(400)+U(400)+P(500)+
S(550)

=2600
bits to
encode

4.) Prof. Gigabyte is correct, the way he stores the Charecters is more space efficcent than the optimal Huffman code.

However the way the Prof stores the Charecters is not feasible for an actual application. Due the ambiguity of the Professors method too many things can go wrong. For example, the binary string 10 can represent AE or D. While the method is space efficient, it is also useless.

↑1

extra Credit