

SPPU-BE-COMP-CONTENT - KSKA Git

DAR

classmate

Date :

Page :

ASSIGNMENT-2

Q1

GREEDY APPROACH:

- Greedy Approach is algorithmic strategy where decisions are made step by step always choosing the option that seems best at the moment (i.e. the locally optimal choice)
- Greedy method works when problem exhibits the greedy choice property & optimal substructure. meaning the local optimum leads to global optimum.

Examples:

1. Huffman coding
2. Kruskal & Prim's Algorithm (For minimum spanning tree)
3. Dijkstra's shortest path Algorithm.

Q2

Huffman coding:

- lossless data compression algorithm that uses greedy approach to assign shorter binary codes to more frequent characters & longer codes to less frequent ones.

SPPU-BE-COMP-CONTENT - KSKA Git

CLASSMATE

Date :

Page :

STEPS:

1. Create Leaf nodes for each character & its frequency.
2. Select 2 smallest frequencies, combine them into node with their sum as the frequency.
3. Repeat until only one node (the root of the Huffman tree) remains.
4. Assign 0 to the left edge & 1 to the right edge to get binary codes.

EXAMPLE

let characters & their freq. be

Character	Frequency
A	5
B	9
C	12
D	13
E	16
F	45

Process

Combine smallest two: $A(5) + B(9) = 14$

Combine $14 + C(12) = 26$

Combine $D(13) + E(16) = 29$

Combine $26 + 29 = 55$

Combine $55 + F(45) = 100$

SPPU-BE-COMP-CONTENT - KSKA Git

Resulting codes

Character	Huffman code.
F	0
C	1 0 0
D	1 0 1
A	1 1 0 0
B	1 1 0 1
E	1 1 1

Hence more freq. characters get shorter codes

Q3

let n be number of unique characters.

- Building the min heap: $O(n)$
- Extracting & Inserting nodes: Each operation takes $O(\log n)$ & there are $(n-1)$ combinations.

Time Complexity

$$O(n \log n)$$

Space Complexity

$$O(n)$$

(for storing tree & the priority queue)