Monday, 8 November 2021 8:36 PM Huffman Coding -> lossless data compression algorithm. The êdea is to use variable length codes to Enput characters ABBCOBCCDAABBEEE BEAB  $A \rightarrow (65) \rightarrow 0100001 = 8 \text{ LHs}.$ Total seg = 20 x 8 = 160 bits. fixed length boding 3 bits -> 8 chor. 8 bits 68 Total bits → 20 k3 = 60 A BB CD - . -000 001 010 ... = 5 x 8 + 5 x 3 = 40 +15 = 55 T - 60 + 55-B B C D B C C D A A B B E E E B E A B freg/court 01 COJ Total bits -> 4x2 + 4x2+ 3x3 + 2x3 + 4x2 8 + 14+9+6+8 = (45) TC -> O (nlogn) Prefix code: code is préfix of another No code. -> David Huffman en 1957 -> encoding follows the grefix rule - most generated char will get the Small code 2 least generated char will get the large code. → TC → o(mlogn) For practise: · char freq 12 16 0 0 100 101 1100 (10) 111

5x8 = 40

COEXOI (code chef) (Jhe unforgetten letters) 2 for (i=n-1 ->0) if (burningtuine 7, a [i]) & if (plag! = 0) & cost = cost + a[i]j burningtime = (burningtime - a[i]); ali]= 1; (burn cost | = 0) { for (t=n-1; izo; i--) } 4f ((a[i]!=-1) 22 (a[i] <= min) && (a[i] 7, (sum-cost))) } min = a[i]; diff = (min + cost) - (sum);