

Hátizsák probléma. ($\Rightarrow$ dinamikus programozás).

Input: - hátizsák S kapacitás

- n tárgy E értékkel és s súllyal. S egész.

Kell: ΣE , ami S kapacitásba belefér.

Részfeladatra bontás: - optimális megoldás kell S kapacitásra.

$enter(x) = \dots$

$enter(0) = 0 < 0$ értéket lehet tenni.

$\uparrow$ 0 kapacitású hátizsákba

futási idő $O(n)$.

$14 + 0 < 1$ kapacitás

$16 + 0$

$9 + 9 - 18$ ez lesz a jó. ismétlődéssel.

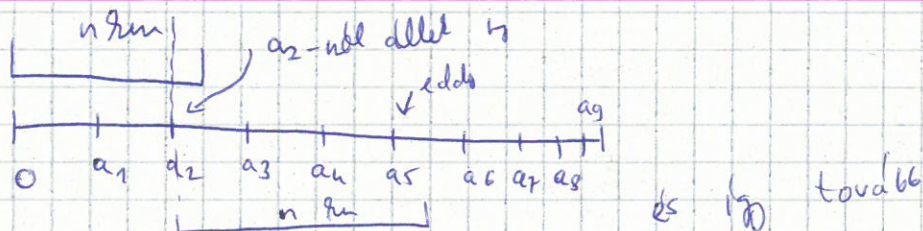
Hátizsák probléma ismétlődéssel és ismétlődés nélkül.

$O(Sn)$ mindig a dűce.

Módszerek algoritmusok:

1. GFT/ly. nem ad optimumot, de hatékony.
2. SFT/ly.
3. 1FT/ly.

Bézsűtás prob. Autóval van, pontosan n km-t tudunk megtenni.



Teljesen n km-t tudunk menni:

0-tól a_8 -ig jussunk el úgy, hogy mind fér.

Σ klasszikus mód.

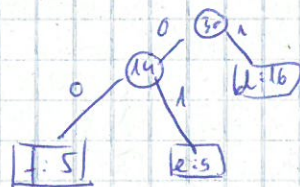
Huffman-rés: vesteselementes tömítés és.

Mi fordul elő többször / Mi fordul elő kevesebb? prefixmentes kódolás és prefix kód fája.

Huffman-algoritmus: bevitel: karakterek + gyakoriság

kimenet: bináris fa

$O(n \log n)$ lép.



... ad?

↑ felépít fel.

1001111 0101101
c d b a e

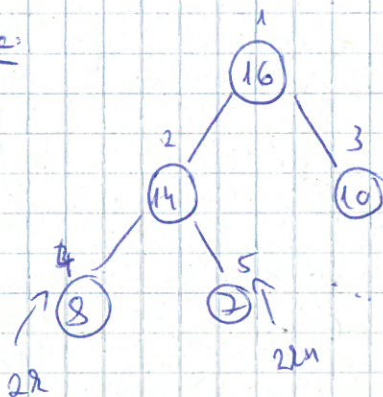
Fa alapján is lehet dekódolni

ZH-n, vizsgáljuk fel kell építeni a Huffman fát.
és dekódolni

Adatszerkezetek (elemek). tömbök, listák, illesztési...

Vélemény és Queue. Kupa, prioritási sa.

Kupa:



tömbben tárolható.

logaritmusos
mértékű.

KIVÉSZ-MAX animáció. fontos

Ittána: Keresőfák.

és működésük.