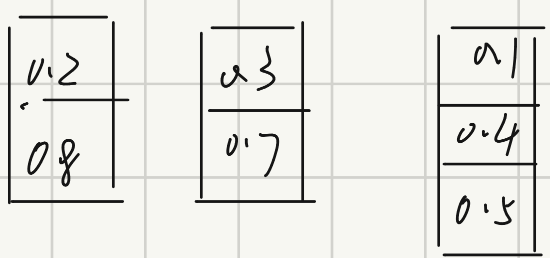
1. all instance.
2. polymial time
3. optimal solution ← Loose this.

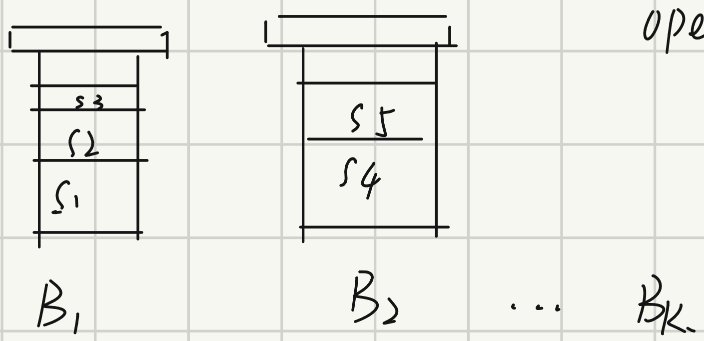
Binpack Problem (NP - hard)

Input: n items with size $S_1, S_2, \cdots, S_n$. $(0 < S_i \le 1)$
Output: packing the items using fewest bins with unit capacity

0.2 0.5 0.4 0.7 0.1 0.3 0.8



Next Fit:



open a bin → fill $\xrightarrow[\text{item can't be added}]{\text{when one more}}$ close bin

$S(B_i)$

$S(B_1) + S(B_2) > 1$

$S(B_2) + S(B_3) > 1$

$\vdots$

$S(B_{K-1}) + S(B_K) > 1$

$S(B_1) + 2S(B_2) + \cdots + 2S(B_{K-1}) + S(B_K) > K-1$

$\sum_{i=1}^{K} S(B_i) > (K-1)/2$

$\Rightarrow$ opt $>$ $(k-1)/2$. (Cause every bin include exactly 1)

$\underset{\text{next fit}}{NF} = K$ $\begin{cases} K = 2m & opt \geq m \\ K = 2m+1 & opt \geq m+1 \end{cases}$

$\Rightarrow$ $NF / opt \leq 2$.

2. approximation algorithm
it has an approximation ratio of (at most) 2.

Given an algorithm A, if for any instance I of a problem,

$\max\left\{A(I) / OPT(I), OPT(I)/A(I)\right\} \leq \rho(|I|)$

we say A is a $\rho(n)$ approximation algorithm

must be 2 ?

$\Rightarrow$ 2n items ; $(\frac{1}{2}, \delta, \frac{1}{2}, \delta, \frac{1}{2}, \delta, \ldots, \frac{1}{2}, \delta)$ $\delta < \frac{1}{n}$

NF: n.

OPT: $\frac{n}{2} + 1$

$\frac{NF}{OPT} = 2 - \frac{4}{n+2}$

First: we get < 2:
Then, we get close to 2,
It follows that the "2" we get is tight.

Any fit:

for $i = 1$ to n.

if any opened bin has enough space.

put item i into one of such bins

else

open a new bin.

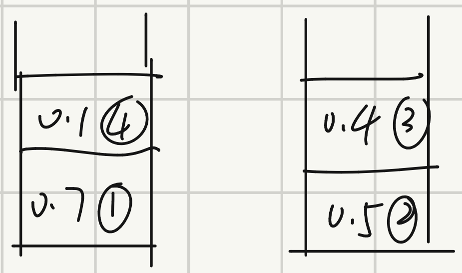
put item i into it

Any fit

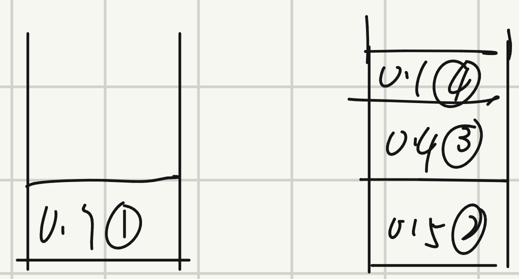
first fit	best fit	worst fit
↓	↓	↓
select first one	select the fullest one	select the most empty one.

0.7 0.5 0.4 0.1

first fit



best fit



Theorem:

absolute approximation ration.

$BF(I)$, $FF(I)$ $\le$ $1.7\,OPT(I)$ for any I, both tight.

$\exists I,\ BF(I)\ (FF(I)) \geq 1.7(OPT(I)-1)$ ⇒ thus tight

proof omitted.

first fit decreasing = sort + first fit
best fit decreasing = sort + best fit.

Theorem:

for any instance I, → asymptotic approximation ration

$FFD(I),\ BFD(I) \leq \frac{11}{9} OPT(I) + \frac{6}{9}$: tight

$$\frac{FFD(I)}{OPT(I)} \leq \frac{\lfloor \frac{11}{9} OPT(I) + \frac{6}{9} \rfloor}{OPT(I)} \leq \frac{3}{2}$$

NF   FF/BF   FFD/BFD → has been proved the best

$\underbrace{2 \quad 1.7}_{\text{Online}}$ $\underbrace{1.5}$

off line (because of sort).

↓

need all input data at one time

Theorem:

For any binpacking problem,
no poly-time algorithm can achieve an approximation ration better than 3/2 unless P = NP.

no online algorithm is better than $\frac{5}{3}$.

L

Knapsack Problem.

Input: n items $(V_1, w_1), \dots, (V_n, W_n)$

capacity C.

Output: fit the knapsack so as to maximize the total value.

$O(nC)$ $\quad O(nV)$ $\quad (V = \sum_i V_i)$

Fraction version: greedy on $\frac{V_i}{W_i}$

Integral version: (NP-hard)

A_1: greedy on $\frac{V_i}{W_i}$

	item	value	weight	
$C=10$	1	2	1	$A_1(I)=2$
	2	9	10	$OPT(I)=9$

A_2: greedy on V_i

	item	value	weight	
$C=10$	$1 \sim 10$	9	1	$A_2(I) = 10$
	11	10	10.	$OPT(I) = 90$.

$A^*(I)$

1. run A_1 and A_2 on I
2. return the better of $A_1(I)$ and $A_2(I)$

Theorem

A^* has an approximation ratio of 2.

Proof:

$$A^*(I) \geq \begin{matrix} A_1(I) \geq OPT_{FRAC}(I) - V_{max} \\ A_2(I) \geq V_{max} \end{matrix}$$

$\Rightarrow$ has a difference smaller than one item, and the item's largest value is V_{max}

Because these two algors

$$2A^*(I) \geq OPT_{FRAC}(I) \geq OPT_{INT}(I)$$

$$\therefore A^*(I) \geq \frac{1}{2} OPT_{INT}(I)$$

$O(nV)$ $\quad V = \sum_i V_i \leq n \cdot V_{max}$

$\hookrightarrow O(n^2 V_{max})$

$\hookrightarrow$ still exponent

Instance: $V_1, \dots V_n$ $\quad d = \gcd(V_1, \dots, V_n)$ $\quad V_1/d, \dots, V_n/d$

$W_1, \dots W_n$. $\quad W_1, \dots, W_n$.

$\Rightarrow$ the same solution.

choose a larger d: $d = \frac{\delta V_{max}}{n}$ $\to$ certain parameter

$\hat{V}_i = \left\lceil \frac{V_i}{d} \right\rceil \to$ scaling

$$\hat{V_{max}} = \lceil \frac{V_{max}}{d} \rceil = \lceil \frac{n}{\delta} \rceil = O(\frac{n}{\delta})$$

$$O(n^2 \hat{V_{max}}) = O(\frac{n^3}{\delta})$$

Since we have rounding for $\hat{v_i}$, so it has difference, the solution changed, different from gcd.

S

$$V(S) = \sum_{i \in S} V_i .$$

$$\underbrace{\hat{V}(S)}_{\text{after scaling}} = \sum_{i \in S} \hat{V_i} = \underbrace{\sum_{i \in S} \lceil \frac{V_i}{d} \rceil}_{\le \sum_{i \in S} (\frac{V_i}{d} + 1) = \frac{V(S)}{d} + |S| = \frac{V(S)}{d} + n} \geq \sum_{i \in S} \frac{V_i}{d} = \frac{V(S)}{d}.$$

$$\Rightarrow V(S) \le d \cdot \hat{V}(S) \le V(S) + nd$$

$$V(S) + \delta V_{max} = V(S) + nd \geq d \cdot \hat{V}(S) \geq V(S)$$

OPT under V_i — Set: S^*, $V(S^*)$

OPT under $\hat{V_i}$ — Set: $\hat{S}$, $V(\hat{S})$

$\Rightarrow$ try to prove $V(S^*)$ is close to $V(\hat{S})$

$$V(\hat{S}) + \delta V_{max} \geq d \hat{V}(\hat{S})$$

$$d \hat{V}(S^*) \geq V(S^*)$$

Because $\hat{S}$ is optimal under $\hat{V}$, S^* is just a right solution $\therefore$ $\hat{V}(\hat{S}) \geq \hat{V}(S^*)$ $\therefore$ $V(\hat{S}) + \delta V_{max} \geq d \hat{V}(\hat{S}) \geq d \hat{V}(S^*) \geq V(S^*)$

$$\begin{cases} V(\hat{S}) + \delta V_{max} \geq V(S^*) \\ V(S^*) \geq V_{max} \end{cases}$$

$$\Rightarrow V(\hat{S}) + \delta V(S^*) \geq V(S^*)$$

$\Rightarrow \quad V(\hat{S}) \geq (1-\delta) V(S^*)$

$$\frac{V(S^*)}{V(\hat{S})} \leq \frac{1}{1-\delta} \leq 1+\varepsilon \qquad (\varepsilon = 2\delta)$$

$O(n^3/\delta) = O\left(\frac{n^3}{\varepsilon}\right)$

δ: changeable. $\delta > 1$

$\Rightarrow$ advantage: ratio can be adjust

A polynomial-time approximation scheme (PTAS) is a family of algorithms $\{A_\varepsilon\}_{\varepsilon>0}$ such that for any $\varepsilon>0$, A_ε is an $(1+\varepsilon)$-approximation algorithm that runs in polynomial in n (given ε is a constant)

$O(n^{\frac{1}{\varepsilon}})$ PTAS

$O(f(\frac{1}{\varepsilon}) \cdot poly(n))$ efficient PTAS

$O(poly(\frac{1}{\varepsilon}) \cdot poly(n))$ fully PTAS $\Rightarrow$ FPTAS

K-center problem.

Input: n site $s_1, \ldots, s_n$ and an integer K.

Output: a set of K centers so as to minimize the maximum distance from a site to its nearest center.

$dist(x,y)$ = distance between x and y.

$$dist(x, C) = \min_{y \in C} dist(x,y)$$

$$r(C) = \max_{x} dist(x, C)$$

find a set C of K center to minimizes $r(C)$

$K=1$

given an algor: select one site as the center

$r^* \geq r/2$.

Assume we know OPT r^* $(0, d_{max})$

while there exists some sites uncovered

pick an arbitrary one as a center

remove all the sites within $2r^*$ from the center

$\log_2 d_{max}$

bisection

$r(C) \leq 2r^*$

assume $|C| \geq K+1$ => But optimal solution has K center, so

$\forall c_i \in C$, $dist(c_i, c_1) > 2r^*$ in the optimal solution,

$\Downarrow$ there exist at least one cover

$\underbrace{r^* \geq}$ $dist(c_i, c_j)/2 > r^*$ → X

$\therefore \quad |C| \leq K$ the two centers in this K+1 solution

Greedy $(s_1, \cdots s_n, k)$

1. $C_1 = \{s_1\}$
2. for $i = 2$ to k.
3. select the site s_j with maximum $dist(s_j, C_{i-1})$
4. $C_i = C_{i-1} \cup \{s_j\}$
5. return C_k

$r(C_k) \leq 2r^*$

Observation $C_K = \{c_1, c_2, \ldots, c_K\}$ r is decreasing

① $r(C_1) \geq r(C_2) \geq \cdots \geq r(C_K)$ With the increasing of centers,

② $C_K = \{a_1, a_2, \ldots, a_K\}$

$dist(a_i, C_{i-1}) = r(C_{i-1}) \geq r(C_K)$

=> $i<j$ $dist(a_i, a_j) \geq dist(a_j, C_{j-1}) \geq r(C_K)$

a_i is in C_{j-1}

Assume $r(C_K) > 2r^*$

k centers

$dist(a_i, a_j) > r(C_K)$

There exist one site x, $dist(x, C) = r(C_K)$

$r^* \geq r(C_K)/2 > r^*$ $\therefore$ $\times$

$\therefore$ $r(C_K) < 2r^*$

Binpacking: from algorithm. $\frac{11}{9}n + \frac{6}{9} \to \frac{3}{2}$.

$B[\frac{2}{3}k]$

① $B[\frac{2}{3}k]$ contains an item with size $> \frac{1}{2}$

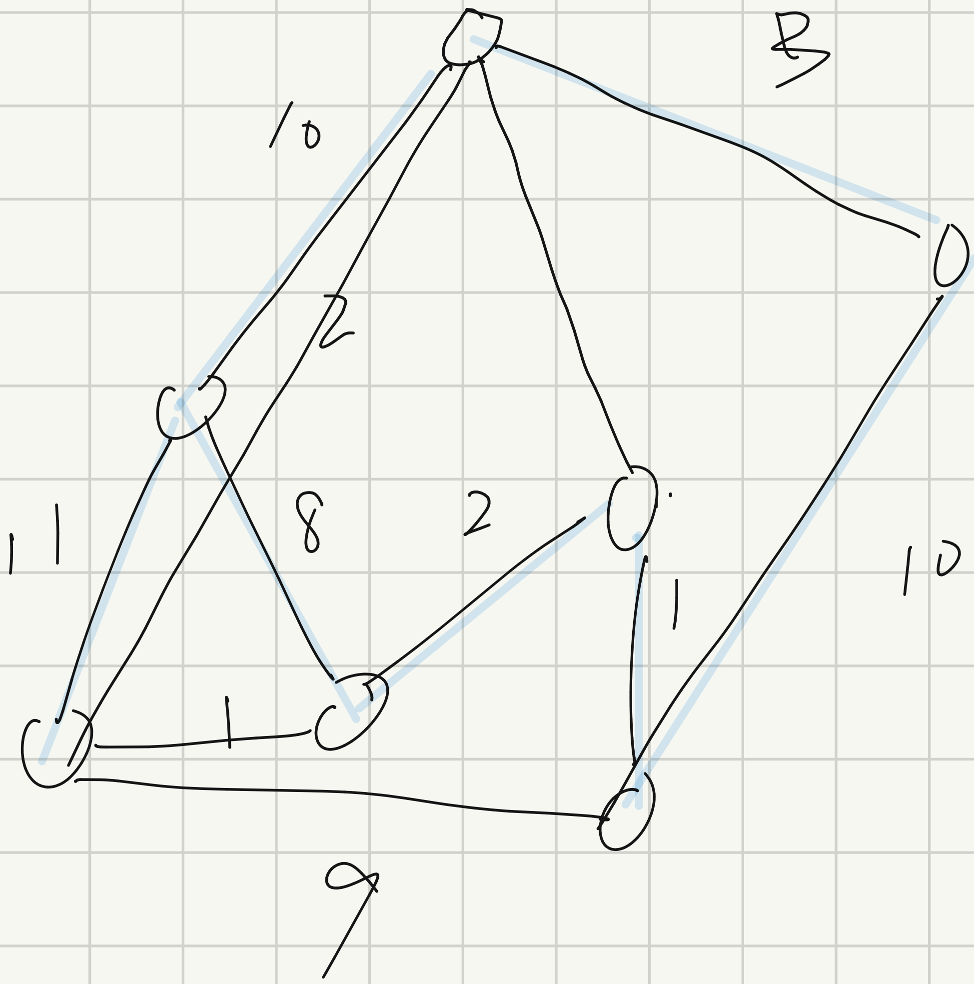
②. every item in $B[\frac{2}{3}k]$ has size $\leq \frac{1}{2}$

$$\frac{FFD(I)}{OPT(I)} \leq \frac{\lfloor \frac{11}{9} OPT(I) + \frac{6}{9} \rfloor}{OPT(I)} \leq \frac{3}{2}$$

$$\underbrace{\frac{1}{2} - 2m\delta, \frac{1}{2} - (2m-1)\delta, \dots, \frac{1}{2} - \delta}_{2m} \qquad \underbrace{\frac{1}{2}+\delta, \frac{1}{2}+2\delta, \dots, \frac{1}{2}+m\delta}_{m}$$

OPT: $m + \frac{1}{2}m$

FFD: $\underline{\qquad m \qquad}$



10 → 11 →