RECAP Optimization version of Subset Sum (determine S^* of maximum $c(S^*) = \sum_{s \in S^*} s$ among all subsets $S' \subseteq S$: $c(S') \leq t$)

EXHAUSTIVE SEARCH IN $\{c(S') \leq t : S' \subseteq S\}$

```
EXP_SS(S,t)
* Let S = {x1, x2, ..., xn} *
L0 = <0>
for i ← 1 to n do
  Li ← MERGE_ND(Li-1, Li-1 +t xi)
  return MAX(Ln)
```

CORRECTNESS: L_i contains all feasible costs of subsets in $S_i = \{x_1, \dots, x_i\}$

RUNNING TIME: In the worst case:

$$\begin{cases} |L_i| = 2|L_{i-1}| \\ |L_0| = 1 \end{cases}$$

$\Rightarrow |L_i| = 2^i$

$\Rightarrow$ The time of MERGE_ND$(L_i, L_i +_t x_i)$ is $\Theta(2^i)$

$n = |S|$

$\Rightarrow T_{ESS}(n) = \Theta(2^n)$!

OBSERVATION: $2^{|S|}$ can be $\Theta(2^{|\langle S, t\rangle|})$!

FROM EXAUSTIVE SEARCH TO FPTAS:

Rather than just eliminating duplicates and values $> t$, we also eliminate costs of subsets that are "well re-presented" by other costs.

Eliminating entries from the L_i's reduces the size of the lists

APPROXIMATION IDEA: If two costs are sufficiently "close" we consider them as "duplicates", and keep only one of them (the smallest of the two to preserve feasibility).

REMARK: "Closeness" will depend on ε!
The smaller ε, the closer the costs must be so that the smaller cost can represent the larger

DEF For $\delta > 0$, a **δ-trimming** of an ordered list L is a sublist L' of L such that:

$$\forall y \in L \ \exists z \in L' : \frac{y}{1+\delta} \le z \le y \qquad y \le z(1+\delta)$$

In a δ-trimming, element $z \in L'$ "is close" to element y. z is smaller than or equal to y but cannot be much smaller! minimo el. di L sarà sempre in L'

Observe that each element $z \in L$ may represent itself (e.g. 0 always represents itself)

EXAMPLE $L = \langle 0, 10, 11, 12, 20, 21 \rangle$, $\delta = \frac{1}{10}$

Then $L' = \langle 0, 10, 12, 20 \rangle$ is a δ-trimming of L, since $\underbrace{\frac{11}{\left(1+\frac{1}{10}\right)}}_{10 =} \le 10 \le 11$ and $\underbrace{\frac{21}{1+\frac{1}{10}}}_{\approx 19.1 =} \le 20 \le 21$

We can obtain the minimum size δ-trimming greedily in linear time:

```
TRIM(L, δ)
* Let L = <y_0, y_1, ..., y_{k-1}>  {|L| = k}
  GC ← y_0  {greedy choice}
  L' ← <GC>
  for i ← 1 to k-1 do
     if (GC < y_i/(1+δ))  {GC not close enough to y_i}
       then
          GC ← y_i
          L' ← <L', GC>
  return L'
```

CORRECTNESS: $\forall y \in L - L' \ \exists z \in L' : y/(1+\delta) \le z \le y$

PROOF: Let $\boxed{y = y_i}$ and let $\boxed{GC_i}$ the greedy choice at iteration i. Clearly, $\boxed{GC_i = y_j}$ for some $\boxed{j < i} \Rightarrow \boxed{y_j = GC_i \leq y = y_i}$. Also, since $\boxed{y = y_i \notin L'}$ it must be $\boxed{GC_i \geq y_i/(1+\delta)}$. The proof follows setting $\boxed{z = GC_i \in L'}$

EXERCISE Prove that $L' = \text{TRIM}(L, \delta)$ is a minimum trimming (standard analysis of a greedy algorithm). However, we will not use minimality directly but the following:

PROPERTY Let $\boxed{L' = \langle z_0 = 0, z_1, \ldots, z_{k-1} \rangle = \text{TRIM}(L, \delta)}$
Then, $\boxed{\text{for } 1 \leq j < k-1: \quad z_{j+1}/z_j > (1+\delta)}$
(The property does not hold for $j = 0$ since $z_0 = 0$)

$\Downarrow$ grande distanza fra el. ⇒ grandezza trimming scende

PROOF z_{j+1} and z_j are two consecutive greedy choices. Thus it must be

$$\boxed{z_j < z_{j+1}/(1+\delta) \Rightarrow z_{j+1}/z_j > (1+\delta)}$$

We are ready to transform EXP-SS$(\langle S, t \rangle)$ into a FPTAS:

(identical to EXP_SS(S,t) but for trimming) with $\delta = \dfrac{\varepsilon}{2n}$

```
APPROX_SS(S, t, ε)
* Let S = {x1, x2, ..., xn} *
L'0 = <0>
for i ← 1 to n do
   L'i ← MERGE_LIS(L'i-1, L'i-1 +t xi)
   L'i ← TRIM(L'i, ε/(2n))
return MAX(L'n)
```

The algorithm is clearly correct, since it returns a feasible cost $\boxed{z' = MAX(L'_n)}$ $(L'_n \subseteq L_n)$

Let $\boxed{z^* = cost(S^*) = \sum_{s \in S^*} s}$. We must prove that $\boxed{\rho_\varepsilon = z^*/z' \leq 1+\varepsilon}$.

We have to study how the trimming error propagates.

LEMMA Let $\boxed{S' \subseteq S_i = \{x_1, x_2, \dots, x_i\}}$ and let $\boxed{y = c(S') = \sum_{s \in S'} s}$. For $\boxed{0 \leq i \leq n}$, let $\boxed{L'_i}$ be the list computed by ASS at the end of iteration i, for $0 \leq i \leq n$. Then

$$\boxed{\exists z \in L'_i : \frac{y}{\left(1+\frac{\varepsilon}{2n}\right)^i} \leq z \leq y}$$

The lemma states that the "closeness" of the representative z gets "looser" exponentially fast in i

PROOF Induction on i. The base $i=1$ is trivial, since $\boxed{L'_1 = \langle 0, x_1 \rangle = L_1}$ (all possible costs in $S_1 = \{x_1\}$). Let $\boxed{i > 1}$ and assume that the HP holds for $i-1$. Two cases:

1. $\boxed{x_i \notin S' \subseteq \{x_1, \dots, x_i\}}$. Then $\boxed{S' \subseteq S_{i-1} = \{x_1, \dots, x_{i-1}\}}$ thus HP holds and for $\boxed{y = c(S') = \sum_{s \in S'} s}$:

$$\boxed{\exists z' \in L'_{i-1} : \frac{y}{(1+\varepsilon/(2n))^{i-1}} \leq z' \leq y}$$

Two subcases:

1.a z' not eliminated in $TRIM(L'_i, \varepsilon/2n)$

Then we have:

$$\boxed{\frac{y}{(1+\varepsilon/2n)^i} \le \frac{y}{(1+\varepsilon/2n)^{i-1}} \le \underset{L'_i}{\underset{\cap}{z'}} \le y}$$

and the theo is follows setting $z = z'$

→ sarà rappresentato

1.b z' is eliminated in $TRIM(L'_i, \varepsilon/2n)$

Then $\exists\, z'' \in L'_i$:

$$\boxed{\boxed{\frac{y}{(1+\varepsilon/2n)^i}} \le \frac{y}{(1+\varepsilon/2n)^{i-1}} \cdot \frac{1}{(1+\varepsilon/2n)} \le \frac{z'}{(1+\varepsilon/2n)} \le \underset{L'_i}{\underset{\cap}{\boxed{z''}}} \le z' \boxed{\le y}}$$

TRIM

and the theo is follows setting $z = z''$

② $\boxed{x_i \in S' \subseteq S_i = \{x_1, \dots, x_i\}}$. Let $\boxed{y = c(S') = \sum_{s \in S'} s}$

We have $\boxed{S' = S'' \cup \{x_i\}}$, with $\boxed{S'' \in S_{i-1}}$,

and $\boxed{y = c(S'') + x_i} \Rightarrow \boxed{c(S'') = y - x_i}$

Then, by HP: $\boxed{\exists\, z' \in L'_{i-1}}$:

$$\boxed{\frac{y - x_i}{(1+\varepsilon/2n)^{i-1}} \le z' \le y - x_i}$$

Thus

$$\boxed{\frac{y}{(1+\varepsilon/2n)^{i-1}} \le \frac{y - x_i}{(1+\varepsilon/2n)^{i-1}} + x_i \le z' + x_i \le y}$$

Observe that $\boxed{z' + x_i \in L'_{i-1} +_t x_i}$.

Again we have two subcases

2.a $z' + x_i \in L'_i \Rightarrow \frac{y}{(1+\varepsilon/(2n))^i} \le z' + x_i \le y$

(reasoning as in 1.a)

2.b $z' + x_i \notin L'_i \Rightarrow \exists z'' \in L'_i$:

$$\frac{y}{(1+\varepsilon/2n)^i} \le z'' \le y$$

(reasoning as in 1.b) ■

Let now $\boxed{y^* = c(S^*) = \sum_{s \in S^*} s \ , \quad S^* \subseteq S = S_n}$

The lemma implies that $\boxed{\exists z' \in L'_n !}$

$$\boxed{\frac{y^*}{(1+\frac{\varepsilon}{2n})^n} \le z' \le y^*}$$

Therefore: $\boxed{\frac{y^*}{z'} \le \left(1+\frac{\varepsilon}{2n}\right)^n}$

OBSERVATION We return $MAX(L'_n) \ge z$

Thus $\boxed{\rho(n) = \frac{c(S^*)}{MAX(L'_n)} \le \frac{y^*}{z'} \le \left(1+\frac{\varepsilon}{2n}\right)^n}$

It suffices to show that $\boxed{\left(1+\frac{\varepsilon}{2n}\right)^n \le (1+\varepsilon) !}$

But:

$\boxed{\frac{d}{dn}\left(1+\frac{\varepsilon}{2n}\right)^n > 0}$ and $\boxed{\lim_{n\to\infty}\left(1+\frac{\varepsilon}{2n}\right)^n = e^{\varepsilon/2}}$

Since $\boxed{e^{\varepsilon/2} \underset{\substack{\Downarrow \\ \text{Taylor}}}{\le} 1+\varepsilon/2+(\varepsilon/2)^2 < 1+\varepsilon/2+\varepsilon/2 = 1+\varepsilon}$

it follows that

$$\boxed{\rho \leq \left(1+\frac{\varepsilon}{2n}\right)^n < \lim_{n \to +\infty} \left(1+\frac{\varepsilon}{2n}\right)^n = e^{\varepsilon/2} < 1+\varepsilon}$$

$\Rightarrow$ A_SS(s,t,ε) is a $(1+\varepsilon)$-approximation!

We are not finished! We need to bound the running time.

Each iteration $i \geq 1$ requires time $\boxed{\Theta(|L'_{i-1}|)}$.

It suffices to limit $\boxed{\sum_{i=1}^{n} |L'_{i-1}| = \sum_{i=0}^{n-1} |L'_i|}$.

Let $\boxed{L'_i = \langle z_0 = 0, z_1, \ldots, z_{k-1} \rangle \quad (|L'_i| = k)}$

We proved that $\boxed{\forall\ 1 \leq j < k-1 : z_{j+1}/z_j > \left(1+\frac{\varepsilon}{2n}\right)}$ (see PROPERTY). We can then write

$$\boxed{\frac{z_{k-1}}{z_1} = \frac{z_{k-1}}{z_{k-2}} \cdot \frac{z_{k-2}}{z_{k-3}} \cdot \ldots \cdot \frac{z_3}{z_2} \cdot \frac{z_2}{z_1} \quad > (1+\varepsilon/2n)}$$

telescoping product

Thus

$$\boxed{\frac{z_{k-1}}{z_1} > \left(1+\frac{\varepsilon}{2n}\right)^{k-2}}$$

Moreover

$$\boxed{\boxed{t} \geq z_{k-1} \boxed{>} \left(1+\frac{\varepsilon}{2n}\right)^{k-2} \cdot z_1 \geq \boxed{\left(1+\frac{\varepsilon}{2n}\right)^{k-2}}}$$

Let us "solve" in k!

$$\boxed{(k-2)\ln\left(1+\frac{\varepsilon}{2n}\right) < \ln t}$$

$$\Rightarrow \boxed{\underset{\substack{\| \\ |L'_i|}}{K} < \frac{\ln t}{\ln\left(1+\frac{\varepsilon}{2n}\right)} + 2}$$

$\rightarrow$ dim. di lista ha sempre ub

However, for $|x|<1$: $\ln(1+x) \geq \frac{x}{1+x}$

$$\Rightarrow |L'_i| = K \leq \frac{\ln t}{\frac{\varepsilon}{2n} \Big/ \left(1+\frac{\varepsilon}{2n}\right)} + 2$$

$$\leq \frac{2n}{\varepsilon} \underbrace{\left(1+\frac{\varepsilon}{2n}\right)}_{\leq 3/2} \ln t + 2$$

$$= O\left(\frac{n}{\varepsilon} \log t\right)$$

Therefore

$$\boxed{T_{ASS} = O\left(\sum_{i=0}^{n-1} |L'_i|\right) = O\left(\frac{n^2 \log t}{\varepsilon}\right)}$$

But $n = O(|<S,t>|)$

and $\log t = O(|<S,t>|)$

Thus $\boxed{T_{ASS}(|<S,t>|, \varepsilon) = O\left(\frac{|<S,t>|^3}{\varepsilon}\right)}$

which is polynomial both in $|<S,t>|$ and $\frac{1}{\varepsilon}$!

EXERCISE: Extend A_SS so that it also returns the subset associated to $\mathrm{MAX}(L'_n)$

RANDOMIZED APPROXIMATION ALGORITHMS

- The use of randomization can help achieve simple and good approximations to hard optimization problems.

A $\boxed{\text{RANDOMIZED ALGORITHM}}$ is an algorithm which may use the primitive RANDOM(S), when $\boxed{S}$ is any finite set.

$X = \text{RANDOM}(S) \in S$ is a random variable returning any element of S with probability $\frac{1}{|S|}$ (uniform distr. over S)

Different runs of RANDOM:

$X = \text{RANDOM}(S_1)$; $Y = \text{RANDOM}(S_2)$

are probabilistically independent, in the sense that

$$\forall x \in S_1, \forall y \in S_2 : \Pr\big((X = x) \wedge (Y = y)\big) = \frac{1}{|S_1|} \cdot \frac{1}{|S_2|}$$

$X \leftarrow \text{RANDOM}(S)$: extraction from urn S with replacement

SPECIAL CASE

$X = \text{RANDOM}(\{0,1\})$ $\left(\Pr(X=0) = \Pr(X=1) = \frac{1}{2}\right)$

is called a COIN FLIP

REMARK the use of RANDOM implies that:

1. The algorithm is not deterministic anymore

⇒ returned value and running time are random variables

2. It may be the case that a randomized algorithm is not always correct

⇒ correctness is an event associated to a probability

Given an optimization problem $\boxed{\Pi}$ a $\boxed{\text{randomized approximation}}$ $\boxed{\text{algorithm}}$ $\boxed{A_\Pi(i)}$ returns a solution $\boxed{s \in S(i)}$ which is a random variable (r.v.) and whose cost $c(s)$ is also a r.v.

DEFINITION

The approximation ratio $\rho(|i|)$ of $A^{k}_{\#}(i) = S_A$ is defined as

$$\rho(|i|) = \max\left\{ \frac{E(c(S_A))}{c(S^*)}, \frac{c(S^*)}{E(c(S_A))} \right\}$$