RECAP:

SET-COVER: Given $\langle X, \mathcal{F} \rangle$, $\mathcal{F} \subseteq \mathcal{P}(X)$ determine min-size covering $\mathcal{C}^* \subseteq \mathcal{F}$:

$$X = \bigcup_{S \in \mathcal{C}^*} S \quad (\text{size} = |\mathcal{C}^*|).$$

```
APPROX-SET-COVER(X, F)
  U ← X ; C ← ∅
  while (U ≠ ∅) do
    S ← argmax{|T ∩ U| : T ∈ F}
    U ← U − S ; C ← C ∪ {S}
  return C
```

We prove a lowerbound on the covering rate:

Let $K = |\mathcal{C}^*|$:

$$\frac{|U_t - U_{t+1}|}{|U_t|} \geq \frac{|U_t|(1/K)}{U_t} = \frac{1}{K}:$$

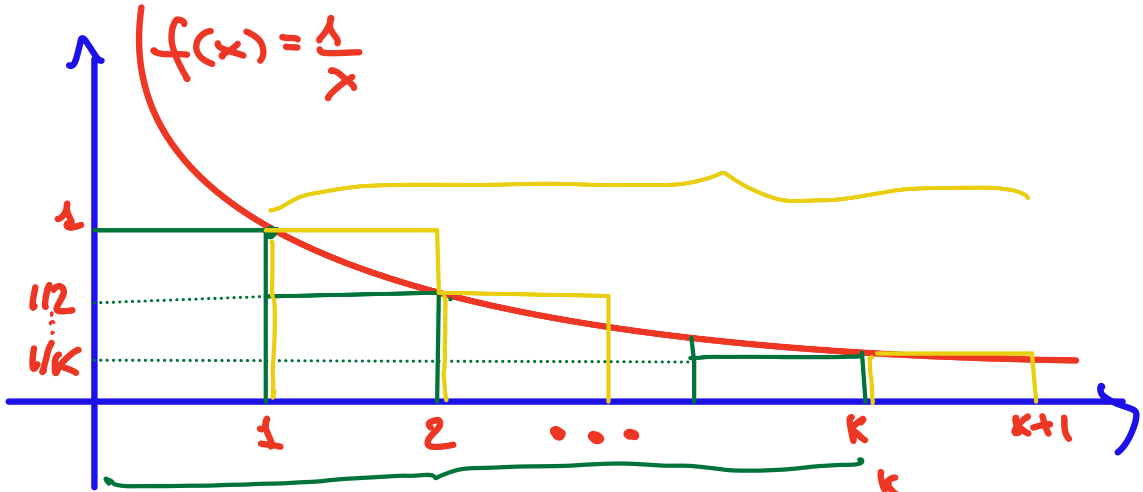
... whence $\rho(n) \leq \lceil \ln n \rceil \leq 1 + \ln n$

TIGHTER (MORE COMPLEX) ANALYSIS

DEF Harmonic number $H(K)$:

$$H(K) = \sum_{i=1}^{K} \frac{1}{i}$$

We have:



$$\sum \square = \sum \square = \sum_{i=1}^{K} \frac{1}{i}$$

Then $\displaystyle\int_1^{K+1} \frac{1}{x}\,dx < \sum_{i=1}^{K} \frac{1}{i} \leq 1 + \int_1^{K} \frac{1}{x}\,dx$

Thus $\ln(K+1) < H(K) \leq 1 + \ln K$

We will prove that given

$|S_{max}| = \max\{|S| : S \in \mathcal{F}\} \leq |X| = n$

$$\boxed{\rho \leq H(|S_{MAX}|) \leq 1 + \ln |S_{MAX}|}$$

REMARK We proved $\rho \le 1 + \ln n$ ($\rho \le \lceil \ln n \rceil$). In the worst case $|S_{MAX}| = \Theta(|X|) = \Theta(n)$! This bound is not better in the worst case but it can be much better for instances with "small" $|S_{MAX}|$.

TECHNIQUE: CHARGING ARGUMENT

Each $x \in X$ is assigned a weight c_x. The weight depends on ASC$(x, \mathcal{F})$ and is "small" if x is covered together with many other elements.

We will relate $|C^*|$ to the weights determined by ASC$(x, \mathcal{F})$ to show that the "covering rate" of the algorithm is good.

RECALL THAT:

$\boxed{\begin{array}{l} U_t: \text{set of elements still to be covered at the beginning of iteration } t \ge 1 \\ S_t: \text{argmax}\{|T \cap U_t| : T \in \mathcal{F}\} \text{ selected subset during iteration } t \end{array}}$

We assign weight c_x to $x \in X$ when x gets covered for the first time at iteration t:

$$\boxed{\text{DEF: Let } x \in S_t \cap U_t \text{ (all elements covered together at iteration } t\text{): Then } c_x = \frac{1}{|S_t \cap U_t|}}$$

REMARK 1: If iteration t covers many elements ($|S_t \cap U_t|$ large) then c_x is small.

REMARK 2: Since A.S.C covers all elements $x \in X$, each x is assigned a given weight c_x, eventually.

PROPERTY 1:

$$\sum_{x \in S_t \cap U_t} c_x = \sum_{x \in S_t \cap U_t} \frac{1}{|S_t \cap U_t|} = \frac{|S_t \cap U_t|}{|S_t \cap U_t|} = 1$$

We assign unit weight at each iteration. Since there are $|\mathcal{C}|$ iterations (where $\mathcal{C}$ is the cover returned by A_S_C):

$$\sum_{x \in X} c_x = \sum_{t=1}^{|\mathcal{C}|} \Big(\underbrace{\sum_{x \in S_t \cap U_t} c_x}_{=1} \Big) = |\mathcal{C}|$$

We are ready to relate $\mathcal{C}^*$ and $\mathcal{C}$ using the weights.

LEMMA

$$\sum_{S \in \mathcal{C}^*} \sum_{x \in S} c_x \geq \sum_{x \in X} c_x = |\mathcal{C}|$$

PROOF: We know that $\mathcal{C}^*$ is a cover. Thus $\forall x \in X$, c_x must be present as a summand in one of the sums on the l.h.s. !

ARGUMENT IDEA: We will prove that

$$\forall S \in \mathcal{F} : \sum_{x \in S} c_x \leq H(|S|)$$

Therefore, from the lemma:

$$|\mathcal{C}| \leq \sum_{S \in \mathcal{C}^*} H(|S|).$$

Let $S_{max} = \arg\max\{|S| : S \in \mathcal{F}\}$

We have:

$$|\mathcal{C}| \leq \sum_{S \in \mathcal{C}^*} H(|S|) \leq |\mathcal{C}^*| H(|S_{max}|)$$

whence:

$$\rho = \frac{|\mathcal{C}|}{|\mathcal{C}^*|} \leq H(|S_{max}|)$$

The only thing to prove to complete the analysis is:

LEMMA

$$\boxed{\forall S \in \mathcal{F} : \sum_{x \in S} c_x \leq H(|S|)}$$

(Intuition: the sum is not too large because A_S_C covers many elements together at each iteration)

PROOF: Let $S \in \mathcal{F}$.

Recall that $S_1, S_2, \ldots, S_{|\mathcal{C}|}$ are the subsets selected by A_S_C and that U_t is the set of uncovered elements before iteration $t \geq 1$

For $1 \leq t \leq |\mathcal{C}|$ define

$$m_t = |S \cap U_{t+1}|$$

The value m_t is the number of elements of S still uncovered after iteration $t \geq 1$

Let $m_0 = |S|$. We have:

$$|S| = m_0 \geq m_1 \geq \dots \geq m_t \geq \dots \geq m_{|\mathcal{C}|} = 0$$

(Observe that it might be $m_{t-1} = m_t$)

Now, in S $m_{t-1} - m_t$ elements are covered at iteration t. The cost c_x of each of these elements is $\dfrac{1}{|S_t \cap U_t|}$ (from definition of c_x)

therefore:

$$\sum_{x \in S} c_x = \sum_{t=1}^{|\mathcal{C}|} (m_{t-1} - m_t) \cdot \frac{1}{|S_t \cap U_t|} \qquad (1)$$

Now, recall that S_t is the greedy choice at iteration t, thus

$$|S_t \cap U_t| \geq |T \cap U_t| \quad \forall T \in \mathcal{F}$$

Therefore:

$$|S_t \cap U_t| \geq |S \cap U_t| = n_{t-1}$$

Substituting in (1):

$$\sum_{x \in S} c_x \leq \sum_{t=1}^{|e|} \frac{n_{t-1} - n_t}{n_{t-1}}$$

From now on, simple manipulations:

$$\sum_{t=1}^{|e|} \frac{n_{t-1} - n_t}{n_{t-1}} = \sum_{t=1}^{|e|} \sum_{j=n_t+1}^{n_{t-1}} \frac{1}{n_{t-1}}$$

CHECK AT HOME!

$$\leq \sum_{t=1}^{|e|} \sum_{j=n_t+1}^{n_{t-1}} \frac{1}{j}$$

$$\leq \sum_{t=1}^{|e|} \left(\sum_{j=1}^{n_{t-1}} \frac{1}{j} - \sum_{j=1}^{n_t} \frac{1}{j} \right) =$$

$$= \sum_{t=1}^{|e|} \left(H(n_{t-1}) - H(n_t) \right) =$$

$$= H(n_0) - H(n_1) + H(n_1) - H(n_2) + \cdots + H(n_{|e|-1}) - H(n_{|e|})$$

(telescoping sum)

$$= H(u_0^{|S|}) - H(u_{|C|}^{=0}) =$$

$$= H(|S|) - H(0) = H(|S|)!$$

OBSERVATIONS

The bound is much better when $|S_{max}|$ is small.

APPLICATION: MINIMUM VERTEX COVER

restriction: graphs of degree ≤ 3:
still hard!
Using the $VC \leq_P SC$ reduction (which preserves the approximation) I obtain

$$\rho_{3\text{-}\Delta\text{-}VC} = H(3) = 1 + \frac{1}{2} + \frac{1}{3} = \frac{11}{6} < 2!$$

An FPTAS for Subset Sum (SS)

Optimization version of SS:

$$\begin{cases} \mathcal{I} = \langle S, t \rangle \; S \subseteq \mathbb{N}^+ \text{ finite};\; t \in \mathbb{N} \\ \mathcal{S} = \{ T \subseteq \mathbb{N}^+ : T \text{ finite} \} \end{cases}$$

Given $i = \langle S, t \rangle : \mathcal{S}(i) = \{ S' \subseteq S : c(S') = \sum_{s \in S'} s \leq t \}$

Type: max

- Determine the subset of S of maximum sum among all those (admissible) whose sum is not larger than target t
- "Best lower approximation" of t

APPLICATION: $S = \{s_1, \dots, s_n\}$ are weights and t is the capacity of the container.
Aim: pack the container as much as possible (hypotheses: ① no volume constraints; ② all items have the same unit value)

HARDNESS: Decision version:

$$\begin{cases} SS' \\ I: \langle S, t, k \rangle \\ Q: \exists S' \subseteq S : \underbrace{k \leq \sum_{s \in S'} s}_{\text{lower bound}} \leq t\,? \end{cases}$$

Clearly $SS <_P SS'$ (generalization)

$$f(\langle S, t \rangle) = \langle S, t, t \rangle$$

The question for SS' under f becomes:

$$\exists S' \subseteq S : t \le \sum_{s \in S'} s \le t$$

$$\Leftrightarrow \exists S' \subseteq S : \sum_{s \in S'} s = t \ ?$$

Same question as SS.

Clearly:

$$\langle S, t \rangle \in SS \Leftrightarrow f(\langle S, t \rangle) = \langle S, t, t \rangle \in SS'$$

We will develop a FPTAS for (the optimization version of) SS : APPROX-SS$(\langle S, t \rangle, \varepsilon)$

$\forall\, 0 < \varepsilon \le 1$, if $S' =$ APPROX-SS(S, t, ε):

$$\text{MAX!} \quad \frac{c(S^*)}{c(S')} = \sum_{s \in S^*} s \Big/ \sum_{s \in S'} s \le (1 + \varepsilon)$$

APPROACH (paradigmatic): we start from an exhaustive approach which enumerates all costs of subsets in $\mathcal{I}(i)$. The exhaustive search is modified (according to ε) to yield the FPTAS.

REMARK Exhaustive search may cost $\Theta(2^{|S|})$

LET $S = \{x_1, x_2, \dots, x_n\}$. W.l.o.g. $\boxed{x_i \le t}$

DEFINE L_i as the ordered list without duplicates of ALL costs of feasible subsets of $S_i = \{x_1, x_2, ---, x_i\}$, $1 \le i \le n$.
Also, define $L_0 = \langle 0 \rangle = \langle c(\emptyset) \rangle$

EXAMPLE For instance $\langle S, 12 \rangle$, with

$$S = \{x_1 = 3, x_2 = 4, x_3 = 7\}$$

$L_0 = \langle 0 \rangle$; $L_1 = \langle 0, 3 \rangle$; $L_2 = \langle 0, 3, 4, 7 \rangle$; $L_3 = \langle 0, 3, 4, 7, 10, 11 \rangle$. Note that

$$x_1 + x_2 + x_3 = 14 \notin L_3$$

DEFINE the operation $L +_t x$ as the list obtained by summing x to all elements of L and removing the values $> t$

e.g.: $L_2 +_{12} 7 = \langle 0+7, 3+7, 4+7, \cancel{7+7} \rangle =$

$$= \langle 7, 10, 11 \rangle$$

clearly: $T_{+_t}(|L|) = \Theta(|L|)$

LET MERGE_WD(A, B) a subroutine doing the merge of two ordered lists A and B which also eliminates duplicates in the resulting list. $T_{MWD}(|A|, |B|) = \Theta(|A| + |B|)$

PROPERTY

For $1 \le i \le n$:

$$L_i = \text{MERGE\_WD}(L_{i-1}, L_{i-1} +_t x_i)$$

PROOF

Let $S \subseteq S_i$. If $x_i \notin S$ then $S \subseteq S_{i-1}$ and $c(S) = \sum_{s \in S} s \in L_{i-1}$. Otherwise:

$$c(S) = \sum_{s \in S - \{x_i\}} s + x_i, \text{ and}$$

$S - \{x_i\} \subseteq S_{i-1}$. Therefore $c(S) \in L_{i-1} +_t x_i$

The ordered list without duplicates L_i can thus be obtained as

$$L_i = \text{MERGE\_WD}(L_{i-1}\ L_{i-1} +_t x_i)$$

We are ready to see the exhaustive algorithm:

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

CORRECTNESS: Straight from PROPERTY: $\text{MAX}(L_n)$ is the maximum cost of a feasible subset

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 $\text{MERGE-UD}(L_i, L_i +_\varepsilon x_i)$ is $\Theta(2^i)$

$n = |S|$

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

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