Section 1.7: Randomized Rounding

Alg RR$_1$

```
Solve LP
I ← ∅
For j ← 1 to m
    With probability x_j
        I ← I ∪ {j}
```

Expected cost $= Z^*_{LP} \leq OPT$, but the result is most likely <u>not</u> a set cover.

Alg RR$_2$

```
Solve LP
  I ← ∅
  For i ← 1 to 2·ln(n)
    For j ← 1 to m
        With probability x_j
            I ← I ∪ {j}
```

Expected cost $\leq 2 \cdot \ln(n) \cdot Z^*_{LP} \leq 2 \cdot \ln(n) \cdot OPT$, and high probability that all elements are covered.

(Calculations below)

Alg RR_3

Solve LP

Repeat

$\quad I \leftarrow \emptyset$

$\quad$ For $i \leftarrow 1$ to $2 \cdot \ln(n)$

$\quad\quad$ For $j \leftarrow 1$ to m

$\quad\quad\quad$ With probability x_j

$\quad\quad\quad\quad I \leftarrow I \cup \{j\}$

Until $\{S_j \mid j \in I\}$ is a set cover and $w(I) \leq 4 \ln(n)\, Z^*_{LP}$

Cost $\leq 4 \cdot \ln(n) \cdot OPT$

Result is a set cover.

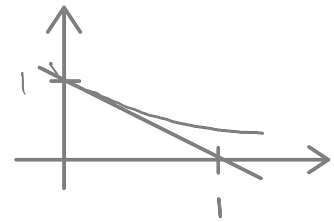
Expected running time is polynomial.

(Calculations below)

p_i : prob. that e_i is covered

$\overline{p_i} = 1 - p_k(i)$: prob. that e_i is <u>not</u> covered

Alg RR_1:



$$\bar{p}_i = \prod_{j: e_i \in S_j} \overbrace{(1-x_j)}^{\le e^{-x_j},} \quad \text{for any } x_j \in \mathbb{R}$$

$$\le \prod_{j: e_i \in S_j} e^{-x_j}$$

$$= e^{\underbrace{-\sum_{j: e_i \in S_j} x_j}_{\le -1, \text{ by the LP constraint corresponding to } e_i}}$$

$$\le e^{-1}$$

Alg RR_2:

$$\bar{p}_i = (\bar{p}_i)^{2\ln n} \le e^{-2\ln n} = \left(e^{-\ln n}\right)^2 = n^{-2}$$

$$Pr[\text{not set cover}] \le \sum_{i=1}^{n} \bar{p}_i \le \sum_{i=1}^{n} n^{-2} = n \cdot n^{-2} = n^{-1}$$

$$\underbrace{Pr[w(I) \geq 4 \cdot \ln(n) \cdot z^*_{LP}]}_{> \frac{1}{2} \text{ would give } E[w(I)] > 2 \cdot \ln(n) \cdot z^*_{LP} \ ↯} \le \frac{1}{2}, \quad \text{by Markov's Inequality:}$$

Alg RR_3:

$$Pr[\text{„not set cover" or „too expensive"}] \le n^{-1} + \frac{1}{2}$$

Thus,

$$E[\#\text{ iterations}] \le \frac{1}{1-(n^{-1}+\frac{1}{2})} \approx 2$$

Sometimes randomized algorithms are simpler / easier to describe / come up with.
Sometimes randomized algorithms can be derandomized as we saw in Chapter 5.
Exercise sheet 7: derandomize $Alg RR_3$ (Ex. 5.7)