LEFTOVER EXCERCISES

1. Useful bounds

a) Prove $n! \leq 2^{n \log_2 n}$.

b) Prove $(1-p)^n \leq e^{-pn}$ for $p \in [0,1]$.

2. Complexity classes

Definitions of the probability complexity classes are on the other side.

a) Show that the following definition is equivalent to the definition of ZPP:

- Always runs in polynomial time.
- If the correct answer is **TRUE** , the algorithm accepts with probability at least $\frac{1}{2}$, otherwise it says "don't know".
- If the correct answer is **FALSE** , the algorithm rejects with probability at least $\frac{1}{2}$, otherwise it says "don't know".

b) Prove the following inclusions:

- P $\subseteq$ ZPP
- RP $\subseteq$ NP
- ZPP $=$ RP $\cap$ co-RP
- RP $\cup$ co-RP $\subseteq$ BPP

LINEAR PROGRAMMING

3. Trading

A trader is trying to make as much money as possible by transporting goods. The budget is 10 000 000 Kč and the maximal weight of the load is 1000 kg. Design a linear program to maximize their profit.

Material	Price per kilo	Profit per kilo
Gold	100 000	100
Silver	14 000	200
Spices	4 000	150
Cocoa	700	10

4. More linear programs

Formulate linear programs for the following problems in a directed graph:

- Shortest s - t path
- Maximal s - t flow
- Minimal s - t cut

Formulate linear programs for the following problems in undirected graphs.

d) Maximum matching

e) Minimal vertex cover

Also determine:

I. What are the relationships between the individual linear programs?

II. What is the relationship between the optimum of the integer program and its relaxations for the programs a), b) a c)?

Hint: Their constraint matrix has a special form with unique properties.

5. Load balancing

Formulate an integer linear program for scheduling / load balancing on identical machines.

OTHER PROBLEMS

6. Parallel coloring

- A Las Vegas algorithm is a randomized algorithm that always answers correctly, but the running time is random for each run and its expected value is finite.

Find a Las Vegas parallel algorithm that uses n processors to properly color a graph with n vertices and maximum degree Δ , using 2Δ colors in expected time $O(\Delta \log n)$.

7. k-Center problem

- We define $R(S) := \max_{v \in V} d(v, S)$, where $d(v, S) := \min_{s \in S} d(v, s)$ for any $S \subset V,\ v \in V$.
- In the k -Center problem:
  - **Input** : We are given a metric space (V, d) and $k \in \mathbb{N}$.
  - **Output** : We want to find $S \subset V$ minimizing $R(S)$ such that $|S| \leq k$.

Find a 2-approximation algorithm the k -Center problem.

Definitions

- A problem is in **BPP** if $\exists$ algorithm satisfying:
  - Always runs in polynomial time.
  - If the correct answer is **TRUE** , algorithms accepts with probability at least $\frac{2}{3}$.
  - If the correct answer is **FALSE** , algorithms accepts with probability at most $\frac{1}{3}$.
- A problem is in **RP** if $\exists$ algorithm satisfying:
  - Always runs in polynomial time.
  - If the correct answer is **TRUE** , algorithms accepts with probability at least $\frac{1}{2}$.
  - If the correct answer is **FALSE** , algorithms always rejects.
- A problem is in **ZPP** if $\exists$ algorithm satisfying:
  - Only the expectation of the running time is polynomial.
  - Always answers correctly.