Problem 1 (5+ 5+ 5+ 5 points):

a) Which of the following are sound measures of algorithmic cost?

- [ ] software licence fee/purchase cost
- [ ] programmers’ salaries
- [ ] runtime (#CPU seconds)
- [ ] runtime (#steps)
- [ ] asymptotic (big-O) #steps
- [ ] asymptotic memory consumption (#bits)
- [ ] asymptotic communication volume (#bits)
- [ ] asymptotic #processors · #parallel steps
- [ ] energy consumption (#kWh)
- [ ] #bugs

b) Which of the following are sound notions of an algorithm’s performance?

- [ ] worst-case
- [ ] best-case
- [ ] typical case
- [ ] average-case†
- [ ] in practice
- [ ] on a benchmark
- [ ] amortized
- [ ] expected (for randomized algorithms)
- [ ] accuracy (for approximation algorithms)
- [ ] competitive (for online algorithms)

c) Explain the differences between (i) a program, (ii) an algorithm, and (iii) a heuristic.

d) Explain the difference between (i) algorithmic cost and (ii) computational complexity.

* Check your multiple choice answers on this paper: ½ point for each correct, 0 for each incorrect
† with respect to a certain probability distribution on the space of inputs...
Problem 2 (5+5+5+5 points):

a) Specify (!) and describe three significantly different algorithms for sorting \( n \) given keys, together with their asymptotic computational worst-case costs. (No proofs required.)

b) Asymptotically analyze the (number of arithmetic operations used by the) high-school method (a.k.a. long multiplication) for calculating, given the coefficients \( a_0, \ldots, a_n \) and \( b_0, \ldots, b_n \) of univariate polynomials \( A(x) = a_0 + a_1x + a_2x^2 + \ldots + a_n x^n \) and \( B(x) = b_0 + b_1x + b_2x^2 + \ldots + b_n x^n \), determine the coefficients \( c_0, \ldots, c_{2n} \) of their product polynomial \( C(x) := A(x) \cdot B(x) \).

c) Verify the correctness of the following formula. Describe a recursive algorithm based on it for the problem from b).

\[
(A_0(x) + A_1(x) x^n) \cdot (B_0(x) + B_1(x) x^n) = C_0(x) + C_1(x) x^n + C_2(x) x^{2n},
\]

where \( C_0(x) := A_0(x) \cdot B_0(x) \), \( C_2(x) := A_1(x) \cdot B_1(x) \), and

\[
C_1(x) := \left( A_0(x) + A_1(x) \right) \cdot \left( B_0(x) + B_1(x) \right) - C_0(x) - C_2(x).
\]

d) Analyze the asymptotic runtime (number of arithmetic operations) of your algorithm from c).

Problem 3 (10x 2 points): Match\(^5\) the algorithms/problems on the left to their least (known) among the classes of asymptotic worst-case runtime/time complexity to the right:

| Binary search among \( n \) sorted elements | \( O(\log^2 n) \) |
| Comparison-based sorting | \( O(\sqrt{n}) \) |
| Connectedness of a given graph | \( O(n) \) |
| Vertex Cover (Problem 5) | \( O(n\log^2 n) \) |

**Edge Cover:** Given a graph \( G=(V,E) \) and \( k \in \mathbb{N} \), do there exist edges \( e_1, \ldots, e_k \in E \) s.t. every vertex \( v \in V \) belongs to some \( e \in \{e_1, \ldots, e_k\} \)?

| Minimum Spanning Tree of a given connected graph with \( n \) vertices and \( O(n) \) edges | \( O(n^2\log^2 n) \) |
| Multiplication of two \( n \times n \) matrices of entries 0,1 | \( O(n^3\log^2 n) \) |
| Syntax test (parsing) w.r.t. a regular grammar | \( P \) |
| Syntax test w.r.t. a context-free grammar | \( \mathcal{NP} \) |
| Searching a given string of length \( n \) for the occurrence of a given substring of length \( O(n) \) | |

\(^5\) Draw your answers on this paper: 2 points for each correct line, 0 for each missing/incorrect one
Problem 4 (5+5+5+5 points):

a) What is the asymptotic (i) worst-case and (ii) amortized cost of incrementing a binary counter, when each bit-flip counts as one step? (No proof is required here…)

b) Prove your second claim from a).

c) Determine the average cost of the following fun algorithm, asymptotically as $n \to \infty$:
Given a tuple $(b_1, \ldots, b_n)$ of $n$ bits, search the (index $j$ of the)
first non-zero bit $b_j$; in case all $b_j$ are zero, count to $2^n-1$ and stop.

d) What is the (i) worst and (ii) expected cost of the following randomized 'algorithm':
Flip a coin. If it comes out heads, stop; otherwise repeat.
Hint: It holds $\sum_n n \cdot p^n = p/(1-p)^2$ for all $|p|<1$.

Problem 5 (5+5+5+5 points): Recall that Vertex Cover is the following optimization problem: Given an undirected graph $G=(V,E)$, find the least number $k=k(G)$ of vertices $v_1, \ldots, v_k \in V$ such that every edge $e \in E$ is incident to (i.e. has among its two end points) at least one vertex from the set $C=\{v_1, \ldots, v_k\}$. The corresponding decision problem asks whether, given $G$ and $\ell$, it holds $k(G) \leq \ell$.

a) Determine $k(G)$ and an optimal Vertex Cover for the following graph $G$:

b) Consider the following greedy algorithm, initialized with $C=\{\}=F$:

WHILE there exists an edge $e=\{a,b\} \in E$,
add $e$ to $F$ and both its end points $a,b$ to $C$
and remove from $E$ all edges incident to $a$ or $b$.

Prove that the resulting set $C$ constitutes a vertex cover of size $2|F| \leq 2 \cdot k(G)$,
i.e. a 2-approximation.

c) Prove that the analysis in b) is optimal by constructing (a family of) graphs $G$
where the above algorithm produces a vertex cover of size $2 \cdot k(G)$.

d) Will the following variant of b) also yield a vertex cover and which approximation ratio?

For each edge $e=\{a,b\} \in E$ add only one (arbitrary) of its end points to $C$
and remove from $E$ all edges incident to that vertex.

Justify your answers!