Algorithmen, KI und Data Science 1 (10-I-AKIDS-1)

Probeklausur, WiSe 2023/2024
8.2.2024

- The exam and all papers on which you wrote during the exam must be returned;
- Use only the provided paper for solutions. You may also use the backside if needed. We will provide additional papers, if needed. Make sure to write your student ID (Matrikelnummer) on every sheet of paper containing your solutions;
- The exam consists of two parts: Part 1 contains 10 multiple-choice questions and Part 2 contains 4 open problems.
- For multiple choice questions (Part 1), simply circle the letter(s) in front of correct answers;
- For the problems in Part 2 all intermediate steps must be clearly written, and not just the final result. The results (even if correct) without intermediate computation steps will not be accepted;
- You may use a scientific calculator (without memorization capabilities);
- The exam is worth 40 points.

  Part 1 (multiple-choice): there can be either 1 or 2 correct answers among the 5 offered answers for each question. Each question carries 2 points: if there is 1 correct answer, selecting only that answer brings you 2 points. If there are 2 correct answers, then selecting each correct one brings 1 point. Selecting an incorrect answers brings a negative 1 point, but the overall evaluation of the question cannot be negative (0 is the minimal number of points you can get for any question).

  Part 2: maximally 5 points for a fully correct solution to each of the four problems.
- You need at least 50% of the maximal points (that is, at least 20 points) to pass the exam.

No.	Problem	Points	Max. points
P1	Multiple Choice		20
P2	Sorting		5
P3	Graph Algorithms		5
P4	State Space Search		5
P5	Machine Learning		5
	Total		40

Matrikelnummer:

Signature:

Part I: Multiple-choice questions

Q1 In some programming language a programming type `short` is used to represent (relatively) small integers. If we have to allocate $N = 2$ *Bytes* of memory for each `short` number, what is the range of integer numbers that we can support?

(a) from -2^{15} to $2^{15} - 1$
(b) from -2 to 2
(c) from -2^{16} to $2^{16} - 1$
(d) from -16 to 15
(e) from -2^{16} to 2^{16}

Q2 Running time of some algorithm, dependent on the size of the input problem n, is given with the function $f(n) = a_3 \cdot n^3 + a_2 \cdot n^2 + a_1 \cdot n + a_0$, with a_0, a_1, a_2, and a_3 as real-valued constants. Which of the following asymptotic bounds for $f(n)$ is correct?

(a) $f(n) = \Theta(n^4)$
(b) $f(n) = O(n^2)$
(c) $f(n) = O(n^4)$
(d) $f(n) = \Theta(n^3)$
(e) $f(n) = \Theta(n^2)$

Q3 Below is the pseudocode of a sorting algorithm known as the "Bubble Sort".

```
bubble_sort(L):
  for i = L.length - 2 downto 0:
     for j = 0 to i:
       if L[j] > L[j+1]:
         tmp = L[j+1]
         L[j+1] = L[j]
         L[j] = tmp
```

Analyze the algorithm and select statements about its properties below that are correct.

(a) Runtime complexity of bubble sort is $T(n) = O(n^2)$, n being the length of the array.
(b) Runtime complexity of bubble sort is $T(n) = O(n)$, n being the length of the array.
(c) Bubble sort performs in place sorting (has constant space complexity).
(d) Runtime complexity of bubble sort is $T(n) = O(n \cdot log\, n)$, n being the length of the array.
(e) Bubble sort does *not* perform in place, it has quadratic space complexity.

Q4 You are creating a hash table with m buckets, with the hash function based on the division method: $h(key) = key\%m$. A collision is when two keys are hashed to the same bucket. Which of the following statements are true:

(a) We can hash keys from 0 to m without any collisions

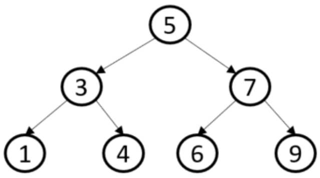
(b) Hashing m^2 different keys will result in m collisions

(c) Hashing $2m$ different keys will result in m collisions

(d) We can hash keys from 0 to $m-1$ without any collisions

(e) We can hash keys from m to m^2 without any collisions

Q5 Which of the statements below hold true for the binary tree in the figure?



(a) This tree satisfies the max-heap property

(b) Converting this tree in a sorted array requires $O(log\, n)$ time, n being the number of elements in the tree (in this case, $n = 7$)

(c) This tree satisfies the min-heap property

(d) This tree satisfies the binary search tree property

(e) Inserting a new key into this tree (and maintaining the property that the tree satisfies) has the time complexity of $O(n)$

Q6 We are solving the state-space-search 8-puzzle problem (image below) with the A* algorithm. We are estimating the remaining cost – the number of moves needed to solve the puzzle – with three different heuristics: h_1 counts the number of misplaced blocks; h_2 sums the Manhattan distances between the current and final position of each block; h_3 always predicts that there are exactly 3 moves left. Which of the following claims about these heuristics are correct?

(a) h_1 is optimistic, but h_2 is pessimistic

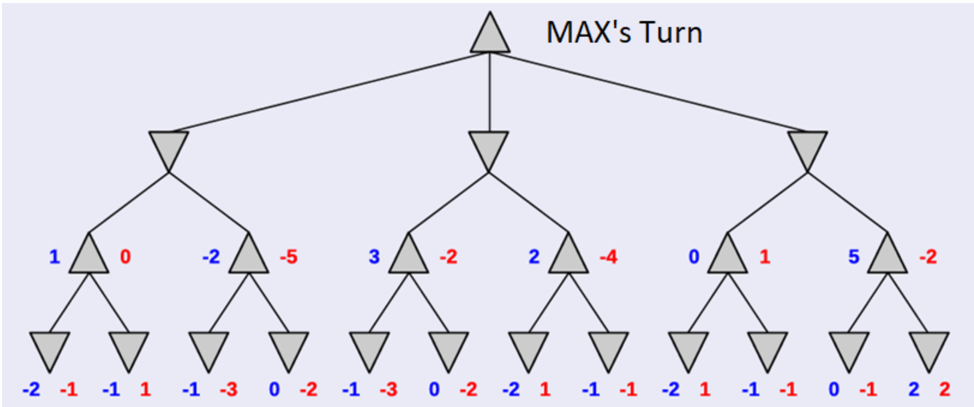
(b) h_1 dominates (is more informed) than h_3

(c) h_3 dominates (is more informed) than h_2

(d) h_2 dominates (is more informed) than h_1

(e) h_3 is optimistic

Q7 A player MAX is playing an adversarial zero-sum game against a player MIN. Both players are using the same strategy: they search the game state tree and select the move that will ultimately lead to the outcome with best utility for them, assuming that their opponent adopts the same strategy. Unfortunately, due to computational limitations, they can each search only **two** moves ahead, and need to make decisions based on heuristic estimates of utility. MAX uses the blue heuristic, MIN uses the red heuristic: both sets of scores (blue and red) are utility estimates for player MAX. If the game is in the state represented by the root node of the game search tree illustrated below, and it's MAX's turn to play, in which state (with what *blue* utility for MAX) will the game end?



(a) -2
(b) 1
(c) 3
(d) 0
(e) -1

Q8 We are solving a traveling salesman problem with $n = 6$ cities using metaheuristic search, namely using a genetic algorithm. In some iteration, the following two chromosomes have been selected for *partially mapped crossover* (with vertical bars indicating cutting points):

$$[1\ 2\ |\ 3\ 4\ |\ 5\ 6] \text{ and } [3\ 5\ |\ 4\ 6\ |\ 2\ 1]$$

Which of the following chromosomes are/is obtained as the result of the crossover?

(a) [1 2 | 4 6 | 5 3]
(b) [1 5 | 4 6 | 2 3]
(c) [3 5 | 4 6 | 5 6]
(d) [6 5 | 3 4 | 2 1]
(e) partially mapped crossover cannot be successfully applied to these two particular chromosomes

Q9 We are applying gradient descent to minimize the following function of two input variables: $f(\mathbf{x} = (x_1, x_2)) = 3x_1^2 - 2x_2^3 + x_1x_2$. We start from the initial point $\mathbf{x}^{(0)} = (x_1^{(0)} = 0, x_2^{(0)} = 1)$ and descent with the step size of $\eta = \frac{1}{6}$. Which of the following is the point $\mathbf{x}^{(1)} = (x_1^{(1)}, x_2^{(1)})$ in which we end after one iteration of gradient descent?

(a) $x_1^{(1)} = -\frac{1}{6}$, $x_2^{(1)} = 0$

(b) $x_1^{(1)} = \frac{1}{6}$, $x_2^{(1)} = 2$

(c) $x_1^{(1)} = 1$, $x_2^{(1)} = -2$

(d) $x_1^{(1)} = -\frac{1}{6}$, $x_2^{(1)} = 2$

(e) $x_1^{(1)} = -\frac{1}{6}$, $x_2^{(1)} = 0$

Q10 We are performing classification (into two classes, + and −) using the k-nearest neighbours (k-NN) algorithm. Our "training" dataset consists of the points below. The inference (i.e., class prediction) is based on Euclidean distance between the points. Which of the following predictions would this classifier make?

Training set: $(\mathbf{x}_1 = [-1,-1], y_1 = +)$, $(\mathbf{x}_2 = [1,1], y_2 = +)$, $(\mathbf{x}_3 = [2,0], y_3 = -)$, $(\mathbf{x}_4 = [-2,0], y_4 = -)$, $(\mathbf{x}_5 = [1,-1], y_5 = +)$, $(\mathbf{x}_6 = [-1,1], y_6 = +)$, $(\mathbf{x}_7 = [0,-2], y_7 = -)$, $(\mathbf{x}_8 = [0,2], y_8 = -)$, $(\mathbf{x}_9 = [0,0], y_9 = -)$

(a) For k = 3 the prediction for $x_{new} = [2,1]$ is −

(b) For k = 3 the prediction for $x_{new} = [2,1]$ is +

(c) This particular training set cannot linearly separate the feature space, so we cannot make predictions for most of the points;

(d) For k = 9 the prediction for any new point is −

(e) For k = 9 the prediction for any new point is +

Part II: Open Questions

P2: Sorting

(a) (**2 pts**) Write the pseudocode of `merge sort`

(b) (**2 pts**) What is the time complexity (worst case runtime) of merge sort? Explain (prove!) your answer.

(c) (**1 pt**) What is the space complexity of merge sort? Explain your answer.

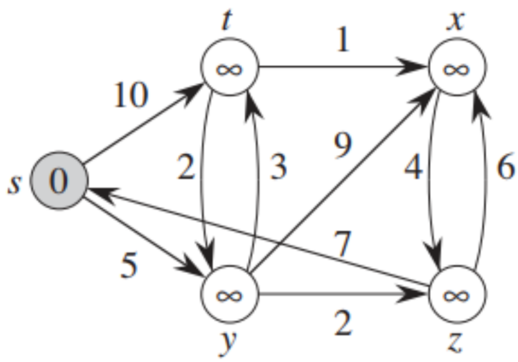
P3: Graphs Algorithms

Both Bellman-Ford and Dijkstra algorithms compute the single-source shortest path distances in a graph: shortest distances of all other nodes from some indicated source node.

(a) (**1 pt**) What is the main difference between Bellman-Ford algorithm and Dijkstra algorithm? Can they be applied on arbitrary graphs?

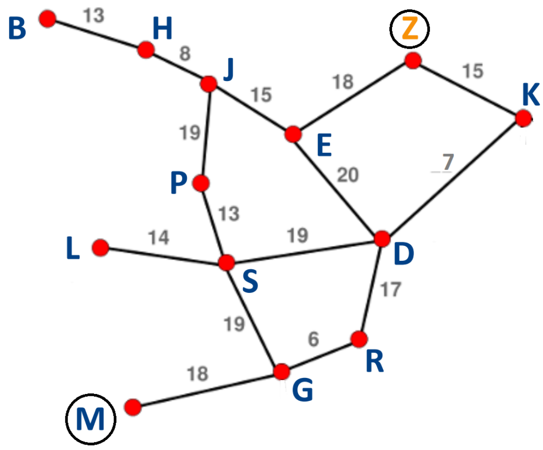
(b) (**2 pts**) What is the runtime complexity of the Bellman-Ford algorithm? Explain

(c) (**2 pt**) Run the Dijkstra algorithm on the graph below, with node s as the source node. In each iteration of the algorithm, clearly indicate all changes in shortest distance of nodes and mark the nodes whose final shortest distance from s has been determined (that is, will certainly not change in subsequent iterations).



P4: State Space Search

We are trying to find the shortest path between the point M and point Z on the map below. Unfortunately, we do not know the distances between points in advance and only realize them after we make the way. This is why we search the space of states using A* algorithm (until we find the goal state Z) (and why we cannot simply run a graph-based algorithm like Dijkstra). We do, however, have estimates of distances from each other point to the goal point Z (based on air distances) which we use as the heuristic h in A*. The node heuristics are given as follows: h(B) = 31, h(D) = 17, h(E) = 12, h(G) = 30, h(H) = 21, h(J) = 17, h(K) = 13, h(L) = 32, h(M) = 40, h(P) = 20, h(R) = 27, h(S) = 25, h(Z) = 0.



(a) **(3 pts)** Execute the A* algorithm step by step and clearly indicate the lists of *visited* and *open* nodes in each iteration.

(b) **(2 pts)** When is a heuristic *consistent*? Is the above heuristic for the map (minimal path from M to Z) consistent?

P5: Machine Learning

We are predicting whether a credit request will be approved (Yes or No) from three features about the applicant: Gender, Age, and Income. You are given the following training set of instances:

Gender	Age	Income	Credit Approved?
Male	Young	High	Yes
Female	Old	Medium	Yes
Non-Binary	Middle-Aged	Low	No
Non-Binary	Young	Medium	No
Non-Binary	Old	Medium	Yes
Male	Old	Medium	No
Female	Middle-Aged	High	Yes
Female	Young	High	No

(a) **(2 pt)** Compute the entropy of the label (Credit Approved?) on this training dataset.

(b) **(3 pt)** Based on the criterion of *information gain* (ID3 algorithm), which of the three features (Gender, Age, or Income) should be selected for the root of the decision tree?