AD2 1D231 Instructions

This is a multiple choice exam. The consists of 4 groups of multiple choice questions. Each question only has one correct answer.

Any assumptions stated in a group only apply to the questions in that group.

The exam is marked out of 20 (1 point for each multiple choice). The grade boundaries are

5	17 .. 20
4	14 .. 16
3	10 .. 13
U	0 .. 9

I will not be able to come to the exam. There is an extra 0p question at the beginning of the exam, where you can add any comments you have about the questions. If any of the questions are incorrect, then you can put a comment there, or if you are unsure about how to answer the question then you can put any assumptions that you make there. I will look at the last question if there are any problems with your exam.

Good Luck!

New Form

If you have any comments on the mulitlpe choice answers that you want me to take into account while marking. Then please fill them in here. You can write something that approximates LaTeX for formulas.

1 Dynamic Programming 8 points Q11

Please see the PDF panel for the questions.

Question 1

- A
- B
- C
- D
- E

Question2

- A
- B
- C
- D
- E

Question 3

- A
- B
- C
- D
- E

Question 4

- A
- B
- C
- D
- E

Question 5

- A
- B
- C
- D
- E

Question 6

- A
- B
- C
- D
- E

Question 7

- A
- B
- C
- D
- E

Question 8

- A
- B
- C
- D
- E

Maximum marks: 8

2 Max Flow 4 points Q8

Question 1

Please select the correct answer

- A
- B
- C
- D
- E

Question 2

Please select the correct answer

- A
- B
- C
- D
- E

Question 3

Please select the correct answer

- A
- B
- C
- D
- E

Question 4

Please select the correct answer

- [ ] A
- [ ] B
- [ ] C
- [ ] D
- [ ] E

Maximum marks: 4

3 Complexity 4 points Q6

Please see the PDF panel for the questions.

Question 1

- A
- B
- C
- D
- E

Question 2

- A
- B
- C
- D
- E

Question 3

- A
- B
- C
- D
- E

Question 4

- A
- B
- C
- D
- E

Maximum marks: 4

4 String Matching 4 points Q3

Question 1

Please select the correct answer

- [ ] A
- [ ] B
- [ ] C
- [ ] D
- [ ] E

Question 2

Please select the correct answer

- [ ] A
- [ ] B
- [ ] C
- [ ] D
- [ ] E

Question 3

Please select the correct answer

- [ ] A
- [ ] B
- [ ] C
- [ ] D
- [ ] E

Question 4

Please select the correct answer

- [ ] A
- [ ] B
- [ ] C
- [ ] D
- [ ] E

Maximum marks: 4

Question 1
Attached

Dynamic Programming

Specification: Given an integer $m \geq 0$ and a set $S = \{s_1, \ldots, s_n\}$ of n integers, with each $s_k > 0$, we want to compute the minimum number of elements of S with sum m.
Example 1: The minimum number of elements of the set $S = \{10, 5, 2, 1\}$ of size $n = 4$ with sum $m = 9$ is 3, for $5 + 2 + 2 = 9$.
Example 2: The minimum number of elements of the set $S = \{5, 4, 3, 1\}$ of size $n = 4$ with sum $m = 7$ is 2, for $4 + 3 = 7$, and not 3, for $5 + 1 + 1 = 7$.
Consider the following Bellman equation — with placeholders α, γ, μ, ψ — for a value $M(i)$:

$$M(i) = \begin{cases} 0 & \text{if } i = 0 \\ \mu\,\{\alpha + M(\gamma) \mid k \in 1\,..\,n \ \wedge\ \psi\} & \text{otherwise} \end{cases}$$

Question 1: If $M(m)$ is returned by a correct algorithm for computing the minimum number of elements of S with sum m, then what is the meaning of $M(i)$, for $i \in 0\,..\,m$?

- $\boxed{\text{A}}$ $M(i)$ denotes the existence of elements of $\{s_1, \ldots, s_i\}$ with sum i
- $\boxed{\text{B}}$ $M(i)$ denotes the minimum number of elements of $\{s_1, \ldots, s_n\}$ with sum i
- $\boxed{\text{C}}$ $M(i)$ denotes the existence of elements of $\{s_1, \ldots, s_i\}$ with sum m
- $\boxed{\text{D}}$ $M(i)$ denotes the number of elements of $\{s_1, \ldots, s_i\}$ with sum i
- $\boxed{\text{E}}$ $M(i)$ denotes the minimum number of elements of $\{s_1, \ldots, s_i\}$ with sum m

Question 2: What is the numeric placeholder α?

$\boxed{\text{A}}$ -1 $\boxed{\text{B}}$ 0 $\boxed{\text{C}}$ 1 $\boxed{\text{D}}$ i $\boxed{\text{E}}$ s_i

Question 3: What is the numeric placeholder γ?

$\boxed{\text{A}}$ $i + s_k$ $\boxed{\text{B}}$ $i - s_k$ $\boxed{\text{C}}$ $\lfloor i/s_k \rfloor$ $\boxed{\text{D}}$ $i - k$ $\boxed{\text{E}}$ $i + k$

Question 4: What is the Boolean placeholder ψ?

$\boxed{\text{A}}$ $s_k \leq m$ $\boxed{\text{B}}$ $s_k < m$ $\boxed{\text{C}}$ $i \geq s_k$ $\boxed{\text{D}}$ true $\boxed{\text{E}}$ $i > s_k$

Question 5: What is the single-argument set-operator placeholder μ?

- $\boxed{\text{A}}$ argmin
- $\boxed{\text{B}}$ set-size greater than or equal to 1

- $\boxed{\text{C}}$ min
- $\boxed{\text{D}}$ set-size
- $\boxed{\text{E}}$ max

Question 6: Which order of computing the $M(i)$ only refers to already computed values?

- $\boxed{\text{A}}$ any order
- $\boxed{\text{B}}$ **for** $i = 0$ **to** m
- $\boxed{\text{C}}$ **for** $i = m$ **downto** 0
- $\boxed{\text{D}}$ **for** $i \in S$
- $\boxed{\text{E}}$ no order

Question 7: Which of the (a) bottom-up iterative and (b) top-down recursive methods for dynamic programming computes every value $M(i)$ (defined by the Bellman equation above) for every i ?

$\boxed{\text{A}}$ only (a) $\boxed{\text{B}}$ only (b) $\boxed{\text{C}}$ both $\boxed{\text{D}}$ neither

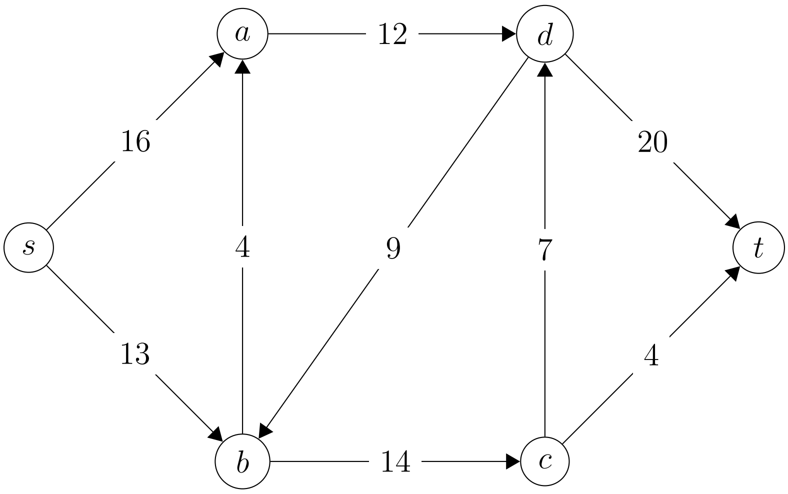
Question 8: Which of the pre-conditions (a) $s_1 \geq s_2 \geq \cdots \geq s_n$ and (b) $s_n \geq s_{n-1} \geq \cdots \geq s_1$ are required for the outlined dynamic program always to terminate correctly?

$\boxed{\text{A}}$ only (a) $\boxed{\text{B}}$ only (b) $\boxed{\text{C}}$ both $\boxed{\text{D}}$ neither

Question 2
Attached

Maximum Flow

Consider the following flow network with source s and sink t:



Question 1: After augmenting along the path $s \to b \to c \to t$, and along $s \to b \to c \to d \to t$, what is the augmenting path of highest capacity?

- [A] $s \to a \to d \to b \to c \to t$: capacity -1
- [B] $s \to a \to d \to b \to c \to t$: capacity 12
- [C] none: reached flow value is maximal
- [D] $s \to a \to d \to t$: capacity 12
- [E] $s \to a \to b \to c \to t$: capacity -2

Question 2: What is the maximum flow value, after all possible augmentations?

- [A] 20
- [B] 24
- [C] 23
- [D] 18
- [E] 29

Question 3: What is the source set S of a minimum cut (S, T)?

- [A] $\{s, a\}$
- [B] $\{s, b\}$
- [C] $\{s, a, b\}$
- [D] $\{s, a, b, c\}$
- [E] $\{s, a, b, d\}$

Question 4: What are the flows across all cuts after all possible augmentations.

- [A] 23
- [B] 17
- [C] 18
- [D] 19
- [E] they differ

Question 3
Attached

Complexity

Question 1: If the best-known *solution checker* for a decision problem D takes $\mathcal{O}(n^k)$ time on an instance of size n, for some $k > 1$, then what is the ***tightest*** time complexity class of D, according to this knowledge?

- [A] NP
- [B] P
- [C] NP-complete
- [D] NP-hard
- [E] none of the others

Question 2: Given an algorithm that computes an assignment of tasks taken from a set $t_1, \ldots, t_n$ based on the integer weights $w_1, \ldots, w_n$ with time complexity $\Theta(\sum_{i=1}^{n} w_i)$: what is the most accurate description of this time complexity?

- [A] logarithmic in $\sum_{i=1}^{n} w_i$
- [B] linear in n
- [C] polynomial in n.
- [D] pseudo-polynomial
- [E] none of the others

Question 3: You have proved that your problem is NP-complete. Which of the following statements is guaranteed true:

- [A] Your problem cannot be solved by pseudo-polynomial time algorithm
- [B] The worst case running time of any algorithm solving your problem takes time exponential time
- [C] It is impossible to find a polynomial time algorithm to solve your problem
- [D] All known algorithms that solve problem have take exponential time in the worst case
- [E] There is a pseudo-polynomial time algorithm that solves you problem

Question 4: In order to prove that a decision problem D is NP-complete, one can:

- [A] prove that D reduces to (often denoted by $\leq_{\mathrm{P}}$) some known problem in P
- [B] prove that some known NP-complete problem reduces to D
- [C] prove that D reduces to some known NP-complete problem and that D is in NP
- [D] prove that some known NP-complete problem reduces to D and that D is in NP
- [E] prove that D reduces to some known NP-complete problem

Question 4
Attached

String Matching

Question 1: On which of the following length-m patterns P does the naïve string matching algorithm reach its ***worst-case*** runtime when looking for ***all*** occurrences of P in the text $T = 0^n$ (that is, a string of n occurrences of the character '0'), with $n \geq m \geq 3$?

$\boxed{\text{A}}$ $0(1^{m-1})$ $\boxed{\text{B}}$ 0^m $\boxed{\text{C}}$ $1(0^{m-1})$ $\boxed{\text{D}}$ $1^{m-1}0$ $\boxed{\text{E}}$ 1^m

Question 2: For the Rabin-Karp string matching algorithm, let p denote the fingerprint of the length-m pattern P, and let t_s denote the fingerprint of the length-m substring T_s for shift s in text T (of length at least m). On which assumption does the algorithm rely? Note that $A \Rightarrow B$ should be read as A logically implies B, and $B \Leftarrow A$ should be read as B is implied by A.

$\boxed{\text{A}}$ $p = t_s \Leftarrow \forall k \in 1\,..\,m : P[k] = T_s[k]$

$\boxed{\text{B}}$ $p = t_s \Leftrightarrow \forall k \in 1\,..\,m : P[k] = T_s[k]$

$\boxed{\text{C}}$ $p = t_s \Rightarrow \forall k \in 1\,..\,m : P[k] = T_s[k]$

$\boxed{\text{D}}$ $p \neq t_s \Leftarrow \exists k \in 1\,..\,m : P[k] \neq T_s[k]$

$\boxed{\text{E}}$ $p \neq t_s \Rightarrow \forall k \in 1\,..\,m : P[k] \neq T_s[k]$

Question 3: How many ***spurious*** hits does the Rabin-Karp string matching algorithm encounter in the text $T =$ "3141512659849792" when looking for ***all*** occurrences of the pattern $P =$ "26", working modulo $q = 11$ and over the alphabet $\Sigma = \{0, 1, 2, \ldots, 9\}$?

$\boxed{\text{A}}$ 0 $\boxed{\text{B}}$ 1 $\boxed{\text{C}}$ 2 $\boxed{\text{D}}$ 3 $\boxed{\text{E}}$ 4

Question 4: On which of the following patterns P does the Rabin-Karp string matching algorithm reach its ***worst-case*** runtime when looking for ***all*** occurrences of P in the text $T = 0^n$ (that is, a string of n occurrences of the character '0'), with $n \geq 3$, working modulo $q = 3$ and over the alphabet $\Sigma = \{0, 1, 2, \ldots, 9\}$?

$\boxed{\text{A}}$ "660" $\boxed{\text{B}}$ "300" $\boxed{\text{C}}$ "099" $\boxed{\text{D}}$ "007" $\boxed{\text{E}}$ "000"