i Instructions AD2

This is a closed book individual exam. You are to work alone and use no resources from the internet. Not treating this as an individual closed book exam is considered cheating.

During the exam I can be contacted via email justin.pearson@it.uu.se
If you have a question then you must tell me the title of your question along with its number.

I will be online 8-9 and 10-11.

The questions are displayed as PDFs. You can use the zoom feature to read magnify the questions. It sometimes helps to resize your browser window.

Good luck.

1 Dynamic Programming 8 points Q2

Question 1

Please choose the correct Answer

- ☐ A
- ☐ B
- ☐ C
- ☐ D
- ☐ E

Question 2

Please choose the correct answer

- ☐ A
- ☐ B
- ☐ C
- ☐ D
- ☐ E

Question 3

Please choose the correct answer

- ☐ A
- ☐ B
- ☐ C
- ☐ D
- ☐ E

Question 4

Please choose the correct answer

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

Question 5

Please choose the correct answer

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

Question 6

Please select the correct answer

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

Question 7

Please choose the correct answer

- A
- B
- C
- D

Question 8

Please choose the correct answer

- A
- B
- C
- D

Maximum marks: 8

2 Max Flow 4 points Q5

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 Disjoint Sets 4 points Q2

Replace with question text

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

4 Complexity 4 points Q1

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

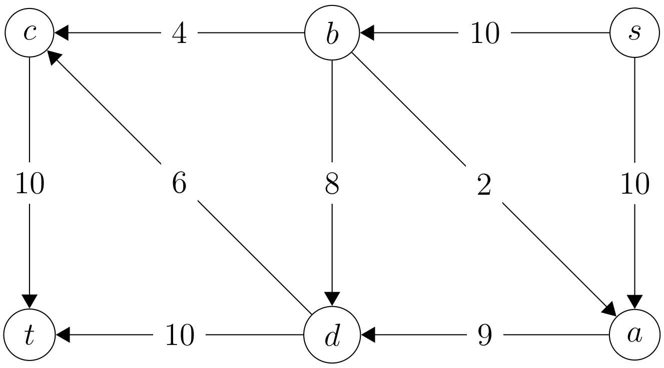
- [ ] A
- [ ] B
- [ ] C
- [ ] D
- [ ] E

Maximum marks: 4

Question 10
Attached

Maximum Flow

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



Question 1: After augmenting along the path $s \to b \to d \to t$, along $s \to b \to a \to d \to t$, and finally along $s \to a \to d \to c \to t$, 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 $+1$
- C none: reached flow value is maximal
- D $s \to a \to b \to c \to t$: capacity -2
- 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 16
- B 17
- C 18
- D 19
- E 20

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

- 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 the 3 augmentations of Question 1?

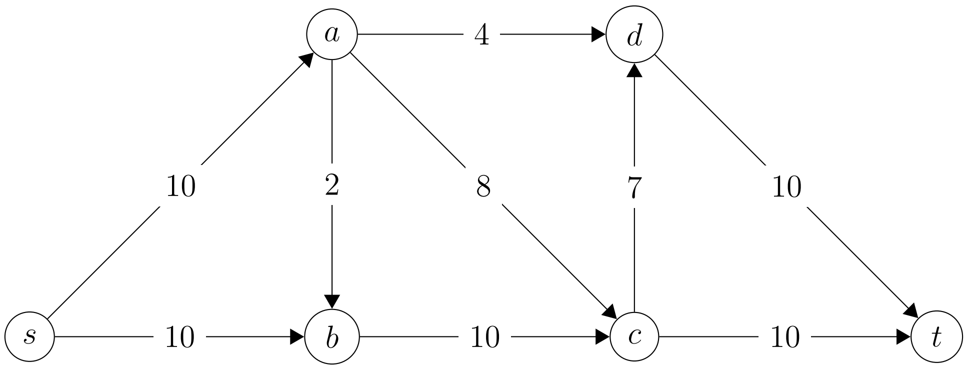
- A 16
- B 17
- C 18
- D 19
- E they differ

Question 11

Attached

Maximum Flow

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



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

- A $s \to b \to c \to a \to d \to t$, capacity +1
- B none, the reached flow value is optimal
- C $s \to b \to c \to d \to t$, capacity +1
- D $s \to b \to a \to d \to t$, capacity +2
- E $s \to b \to a \to c \to t$, capacity +2

Question 2: Are the flows across ***all*** cuts after the 3 augmentations of Question 1 equal?

A yes: 16 B yes: 17 C yes: 18 D yes: 19 E no

Question 3: What is the maximum flow value (after ***all*** possible augmentations)?

A 16 B 17 C 18 D 19 E 20

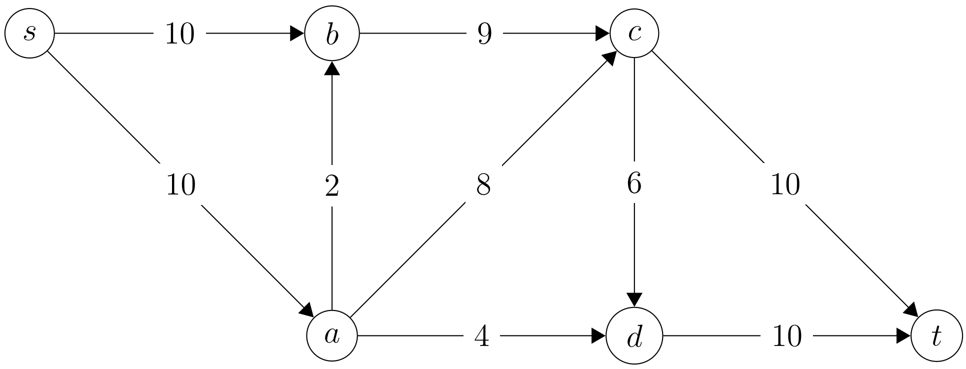
Question 4: What is the capacity of a minimum (s,t)-cut?

A 16 B 17 C 18 D 19 E 20

Question 12
Attached

Maximum Flow

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



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

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

Question 2: Are the flows across all cuts after the 3 augmentations of Question 1 equal?

[A] yes: 16 [B] yes: 17 [C] yes: 18 [D] yes: 19 [E] no

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

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

Question 4: What is the capacity of a minimum cut?

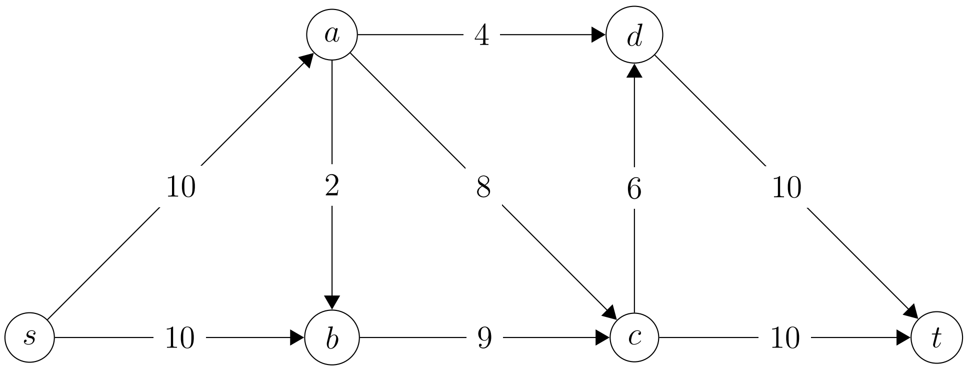
[A] 16 [B] 17 [C] 18 [D] 19 [E] 20

Question 13

Attached

Maximum Flow

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



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

- A $s \to b \to c \to a \to d \to t$, capacity -1
- B none, the reached flow value is optimal
- C $s \to b \to c \to a \to d \to t$, capacity $+1$
- D $s \to b \to a \to d \to t$, capacity -2
- E $s \to b \to a \to d \to t$, capacity $+2$

Question 2: Are the flows across ***all*** cuts after the 3 augmentations of Question 1 equal?

A yes: 16   B yes: 17   C yes: 18   D yes: 19   E no

Question 3: What is the maximum flow value (after ***all*** possible augmentations)?

A 16   B 17   C 18   D 19   E 20

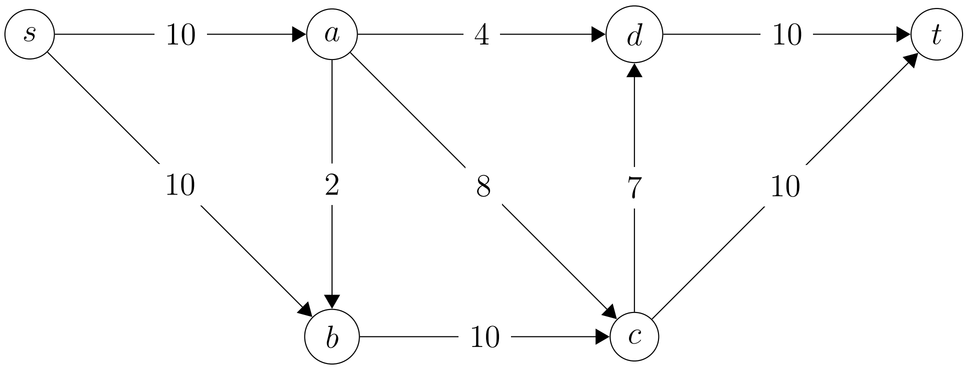
Question 4: What is the source set S of a minimum (s,t)-cut (S,T)? (It is unique here.)

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

Question 14
Attached

Maximum Flow

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



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

- A $s \to b \to a \to d \to t$, capacity 2
- B none, the reached flow value is optimal
- C $s \to b \to a \to c \to t$, capacity 2
- D $s \to b \to c \to d \to t$, capacity 1
- E $s \to b \to c \to a \to d \to t$, capacity 1

Question 2: Are the flows across ***all*** cuts after the 3 augmentations of Question 1 equal?

A yes: 16 B yes: 17 C yes: 18 D yes: 19 E no

Question 3: What is the maximum flow value (after ***all*** possible augmentations)?

A 16 B 17 C 18 D 19 E 20

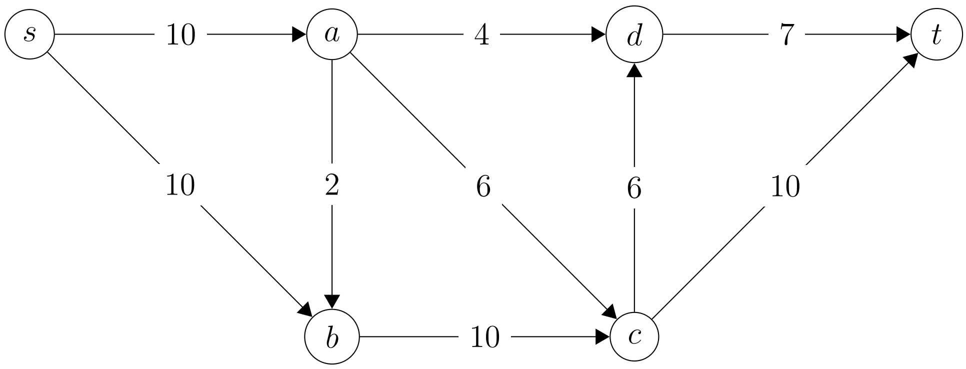
Question 4: What is the capacity of a minimum-capacity (s, t)-cut?

A 16 B 17 C 18 D 19 E 20

Question 15
Attached

Maximum Flow

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



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

- A $s \to b \to a \to d \to t$, capacity 1
- B none, the reached flow value is optimal
- C $s \to b \to c \to t$, capacity 2
- D $s \to b \to c \to d \to t$, capacity 1
- E $s \to b \to c \to a \to d \to t$, capacity 1

Question 2: Are the flows across ***all*** cuts after the 3 augmentations of Question 1 equal and to which value?

A yes: 14 B yes: 15 C yes: 17 D yes: 18 E no

Question 3: What is the maximum flow value (after ***all*** possible augmentations)?

A 16 B 17 C 18 D 19 E 20

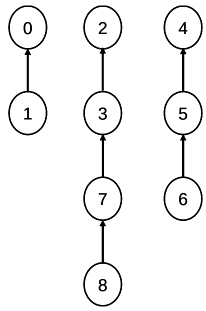
Question 4: What is the capacity of a minimum-capacity (s, t)-cut?

A 16 B 17 C 18 D 19 E 20

Question 16
Attached

Disjoint Sets

Consider the forest on the right, representing disjoint sets for the integers 0 to 8. Assume the rank of each tree currently is its height, measured as a number of arcs. Assume the FIND-SET operation performs path compression. Assume the UNION operation follows the union-by-rank strategy. (Ignore the fact that this forest cannot be obtained using this setup when starting from singleton sets.) Answer the following questions, always starting from the current situation described above:



Question 1: Which element is the parent of element 0 after $\text{UNION}(5, 1)$?

[A] 0, a root [B] 1 [C] 2 [D] 4 [E] other

Question 2: What is the rank of the tree with element 6 after $\text{UNION}(0, 4)$?

[A] 0 [B] 1 [C] 2 [D] 3 [E] other

Question 3: Which element is the parent of element 6 after $\text{FIND-SET}(6)$?

[A] 0 [B] 1 [C] 4 [D] 5 [E] 6, a root

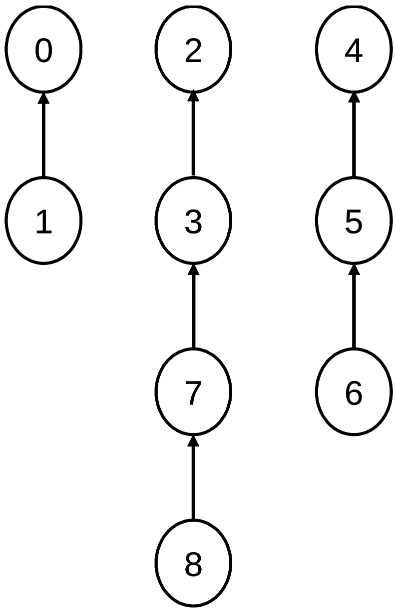
Question 4: What is the rank of the tree with element 5 after $\text{FIND-SET}(6)$?

[A] 0 [B] 1 [C] 2 [D] 3 [E] other

Question 17
Attached

Disjoint Sets

Consider the forest below, representing disjoint sets for the integers 0 to 8. Assume the ranks of the trees ***initially*** are ***equal*** to their heights, measured as numbers of arcs. Assume the FIND-SET operation performs ***path compression***. Assume the UNION operation follows the ***union-by-rank*** strategy. (Ignore the fact that this forest cannot be obtained using this UNION operation when starting from singleton sets.) Answer the following questions, ***continuing at each question from the result of performing the operation of the previous question***:



Question 1: Which element is the parent of element 0 after $\text{UNION}(0, 2)$?

[A] 0 is a root [B] 1 [C] 2 [D] 3 [E] another element

Question 2: What is the rank of the tree with element 4 after $\text{UNION}(2, 4)$?

[A] 0 [B] 1 [C] 2 [D] 3 [E] another value

Question 3: Which element is the parent of element 6 after $\text{FIND-SET}(6)$?

A 6 is a root

B 2

C 4

D 5

E another element

Question 4: What is the rank of the tree with element 8 after FIND-SET(8)?

A 0

B 1

C 2

D 3

E another value

Question 18
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"

Question 19

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 $n \geq m \geq 3$?

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

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?

$\boxed{\text{A}}$ $p = t_s \Rightarrow P = T_s$

$\boxed{\text{B}}$ $p = t_s \Leftrightarrow P = T_s$

$\boxed{\text{C}}$ $p = t_s \textbf{if} P = T_s$

$\boxed{\text{D}}$ $p \neq t_s \textbf{if} \exists k \in 1 \ldots m : P[k] \neq T_s[k]$

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

Question 3: How many spurious hits does the Rabin-Karp string matching algorithm encounter in the text $T =$ "3141512653849792" 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 ***best***-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"

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 amount of elements of S with sum m.
Example 1: The minimum amount 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 amount 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 $M(i)$:

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

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

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

Question 2: What is the numeric placeholder α?

A -1 B 0 C 1 D $\lfloor i/s_k \rfloor$ E $\lceil i/s_k \rceil$

Question 3: What is the numeric placeholder γ?

A $i \div s_k$ B $i - s_k$ C $\lfloor i/s_k \rfloor$ D $i \div k$ E $i - k$

Question 4: What is the Boolean placeholder ψ?

A $s_k < m$ B $s_k \leq m$ C $i \geq s_k$ D $i > s_k$ E $i \leq s_k$

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

A min B sum C set-size D argmin E max

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

A any order B **for** $i = 0$ **to** n C **for** $i = n$ **downto** 0 D **for** $i = 0$ **to** m E **for** $i = m$ **downto** 0

Question 7: Which of the pre-conditions (a) $s_1 > s_2 > \cdots > s_n$ and (b) $s_n = 1$ 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 8: Which of the (a) bottom-up iterative and (b) top-down recursive methods for dynamic programming computes each value defined by the Bellman equation above?

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

Question 20
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^m $\boxed{\text{B}}$ $0(1^{m-1})$ $\boxed{\text{C}}$ $1(0^{m-1})$ $\boxed{\text{D}}$ $1^{m-1}0$ $\boxed{\text{E}}$ 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?

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

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

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

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

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

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', ..., '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', ..., '9'$\}$?

$\boxed{\text{A}}$ "660" $\boxed{\text{B}}$ "300" $\boxed{\text{C}}$ "099" $\boxed{\text{D}}$ "007" $\boxed{\text{E}}$ "006"

Question 21
Attached

Complexity

Question 1: There is an algorithm that computes the power a^n of a given number a for a given natural number n in $\Theta(\log_2 n)$ time: what is the most accurate description, in terms of n, of this time complexity?

- $\boxed{\text{A}}$ logarithmic
- $\boxed{\text{B}}$ linear
- $\boxed{\text{C}}$ pseudo-polynomial
- $\boxed{\text{D}}$ super-exponential
- $\boxed{\text{E}}$ none of the others

Question 2: The classical algorithm for computing naïvely (without knowledge of arithmetic progressions) the factorial $n!$ of a given natural number n takes $\Theta(n)$ time: what is the most accurate description of this time complexity?

- $\boxed{\text{A}}$ logarithmic
- $\boxed{\text{B}}$ linear
- $\boxed{\text{C}}$ pseudo-polynomial
- $\boxed{\text{D}}$ super-exponential
- $\boxed{\text{E}}$ we do not know

Question 3: To the best of our knowledge a decision problem is in NP if and only if its answer takes ...

- $\boxed{\text{A}}$ ... non-polynomial time to check the correctness of a solution.
- $\boxed{\text{B}}$ ... non-polynomial time to compute.
- $\boxed{\text{C}}$ ... polynomial time to check to check the correctness of a solution.
- $\boxed{\text{D}}$ ... polynomial time to compute.
- $\boxed{\text{E}}$ ... possibly forever to compute.

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

- $\boxed{\text{A}}$ prove that D reduces to (denoted by $\leq_{\text{P}}$) some known problem in P
- $\boxed{\text{B}}$ prove that D reduces to some known NP-complete problem

$\boxed{\text{C}}$ prove that D reduces to some known NP-complete problem and that D is in NP

$\boxed{\text{D}}$ prove that some known NP-complete problem reduces to D

$\boxed{\text{E}}$ prove that some known NP-complete problem reduces to D and that D is in NP

Question 22

Attached

Complexity

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

$\boxed{\text{A}}$ P

$\boxed{\text{B}}$ NP

$\boxed{\text{C}}$ NP-complete

$\boxed{\text{D}}$ NP-hard

$\boxed{\text{E}}$ none of the others

Question 2: There is an algorithm that generates the list $[0, 1, 2, \ldots, n-1]$ for a given natural number n in $\Theta(n)$ time: what is the most accurate description of this time complexity?

$\boxed{\text{A}}$ loga-rithmic

$\boxed{\text{B}}$ linear

$\boxed{\text{C}}$ pseudo-polynomial

$\boxed{\text{D}}$ super-exponential

$\boxed{\text{E}}$ none of the others

Question 3: There is an algorithm that computes the power a^n of a given number a for a given natural number n in $\Theta(\log_2 n)$ time: what is the most accurate description of this time complexity?

$\boxed{\text{A}}$ logarithmic

$\boxed{\text{B}}$ linear

$\boxed{\text{C}}$ pseudo-polynomial

$\boxed{\text{D}}$ super-exponential

$\boxed{\text{E}}$ none of the others

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

$\boxed{\text{A}}$ prove that D reduces to (often denoted by $\leq_{\text{P}}$) some known problem in P

- $\boxed{\text{B}}$ prove that D reduces to some known NP-complete problem
- $\boxed{\text{C}}$ prove that D reduces to some known NP-complete problem and that D is in NP
- $\boxed{\text{D}}$ prove that some known NP-complete problem reduces to D and that D is in NP
- $\boxed{\text{E}}$ prove that some known NP-complete problem reduces to D

Question 23
Attached

Complexity

Question 1: If the best known algorithm for solving a decision problem Q takes $\mathcal{O}(k^n)$ time on an instance of size n, for a constant $k > 1$, then what is the ***tightest*** complexity class of Q, according to current knowledge?

- A P
- B NP
- C NP-complete
- D NP-hard
- E we do not know

Question 2: If the best known solution checker for a decision problem Q takes $\mathcal{O}(n^k)$ time on an instance of size n, for a constant $k > 0$, then what is the ***tightest*** complexity class of Q, according to current knowledge?

- A P
- B NP
- C NP-complete
- D NP-hard
- E we do not know

Question 3: The classical algorithm for computing the factorial $n!$ of a given natural number n takes $\Theta(n)$ time: what is the most accurate description of this time complexity?

- A logarithmic
- B linear
- C pseudo-polynomial
- D super-exponential
- E we do not know

Question 4: Which of the following statements are true

- A Finding the longest simple path from a given source to a given target in a graph can be solved in polynomial time.

$\boxed{\text{B}}$ Finding the longest simple path from a given source to a given target in a graph can be converted into a shortest path problem by considering the complement of the input graph.

$\boxed{\text{C}}$ Finding the longest simple path from a given source to a given target in a graph is NP-complete.

$\boxed{\text{D}}$ Finding the shortest path from a given source to a given target is *not* in the complexity class NP.

$\boxed{\text{E}}$ There is a dynamic programming algorithm that finds the longest simple path from a given source that runs in polynomial time and space.

Question 24
Attached

Complexity

Question 1: To the best of our knowledge a decision problem is in NP if and only if its answer takes ...

- $\boxed{\text{A}}$ ... non-polynomial time to check.
- $\boxed{\text{B}}$ ... non-polynomial time to compute.
- $\boxed{\text{C}}$ ... polynomial time to check.
- $\boxed{\text{D}}$ ... polynomial time to compute.
- $\boxed{\text{E}}$ ... possibly forever to compute.

Question 2: To the best of our knowledge a decision problem is NP-hard if and only if ...

- $\boxed{\text{A}}$ ... it can be reduced in polynomial time to every problem in NP.
- $\boxed{\text{B}}$ ... it can be reduced in polynomial time to every NP-complete problem.
- $\boxed{\text{C}}$ ... every problem in P can be reduced in polynomial time to it.
- $\boxed{\text{D}}$ ... every problem in NP can be reduced in polynomial time to it.
- $\boxed{\text{E}}$ ... every NP-complete problem can be reduced in polynomial time to it.

Question 3: If the best known solution checker for a decision problem D takes $\mathcal{O}(n^{k^2})$ time on an instance of size n, for a constant $k > 0$, then what is the ***tightest*** complexity class of D, according to current knowledge?

- $\boxed{\text{A}}$ P
- $\boxed{\text{B}}$ NP
- $\boxed{\text{C}}$ NP-complete
- $\boxed{\text{D}}$ NP-hard
- $\boxed{\text{E}}$ we do not know

Question 4: Given an algorithm that is implemented using dynamic programming which of the following statements best describes the complexity of the implementation.

- $\boxed{\text{A}}$ Always polynomial time
- $\boxed{\text{B}}$ Always NP-complete
- $\boxed{\text{C}}$ Always pseudo polynomial
- $\boxed{\text{D}}$ $\mathcal{O}(n^2)$
- $\boxed{\text{E}}$ we do not have enough information

Question 2
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 amount of elements of S with sum m.
Example 1: The minimum amount 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 amount 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 amount of elements of S with sum m, then what is the meaning of $M(i)$, for $i \in 0\,..\,m$?

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

Question 2: What is the numeric placeholder α?

A -1 B 0 C 1 D $\lfloor i/s_k \rfloor$ E $\lceil i/s_k \rceil$

Question 3: What is the numeric placeholder γ?

A $i - s_k$ B $i + s_k$ C $\lfloor i/s_k \rfloor$ D $i - k$ E $i + k$

Question 4: What is the Boolean placeholder ψ?

A $s_k \leq m$ B $s_k < m$ C true   D $i > s_k$ E $i \geq s_k$

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

A argmin   B average   C set-size   D min   E max

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

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

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

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

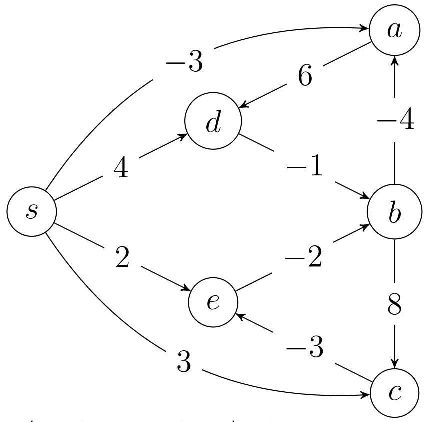
Question 8: Which of the pre-conditions (a) $s_1 > s_2 > \cdots > s_n$ and (b) $s_n = 1$ are required for the outlined dynamic program always to terminate correctly?

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

Question 3
Attached

Dynamic Programming

Consider a weighted directed graph G with a set V of n vertices, a set E of m edges, and an edge-weight function $w\colon E \to \mathbb{R}$, such as in the figure to the right.



Consider the following recurrence, parameterised by $\langle \alpha, \beta_1, \ldots, \beta_4, \gamma \rangle$, for a quantity $T[i, v]$:

$$T[i, v] = \begin{cases} 0 & \text{if } \alpha \wedge (v = s) \\ \beta_1 & \text{if } \alpha \wedge (v \neq s) \\ \gamma\left(T[i-1, \beta_2],\ \min\limits_{(x,v)\in E}(T[i-1, \beta_3] + \beta_4)\right) & \text{if } \neg\alpha \end{cases}$$

Question 1: If $T[n-1, v]$ is returned by a correct algorithm for computing the weight of a shortest path in G from s to v, then $T[i, v]$, with $0 \leq i < n$ and $v \in V$, denotes the weight of a shortest path from s to v with how many edges?

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

Question 2: For the graph given above, what is the sum $T[3, a] + T[3, b]$?

$\boxed{\text{A}}$ -8 $\boxed{\text{B}}$ -6 $\boxed{\text{C}}$ -5 $\boxed{\text{D}}$ -4 $\boxed{\text{E}}$ -3

Question 3: What is the logical condition α?

$\boxed{\text{A}}$ $i = 0$ $\boxed{\text{B}}$ $i = n-1$ $\boxed{\text{C}}$ $i = m-1$ $\boxed{\text{D}}$ $i = m$ $\boxed{\text{E}}$ $i \cdot v = 0$

Question 4: What is the numeric expression β_1?

$\boxed{\text{A}}$ $-\infty$ $\boxed{\text{B}}$ -1 $\boxed{\text{C}}$ 0 $\boxed{\text{D}}$ $+1$ $\boxed{\text{E}}$ $+\infty$

Question 5: What is the index expression β_2?

$\boxed{\text{A}}$ s $\boxed{\text{B}}$ $v-1$ $\boxed{\text{C}}$ v $\boxed{\text{D}}$ $v+1$ $\boxed{\text{E}}$ x

Question 6: What is the index expression β_3?

$\boxed{\text{A}}$ s $\boxed{\text{B}}$ $v-1$ $\boxed{\text{C}}$ v $\boxed{\text{D}}$ $v+1$ $\boxed{\text{E}}$ x

Question 7: What is the numeric expression β_4?

$\boxed{\text{A}}$ $w(s,v)$ $\boxed{\text{B}}$ $w(s,x)$ $\boxed{\text{C}}$ $w(v,x)$ $\boxed{\text{D}}$ $w(x,v)$ $\boxed{\text{E}}$ 1

Question 8: What is the two-argument operator γ (written in prefix form above)?

$\boxed{\text{A}}$ $+$ $\boxed{\text{B}}$ Σ $\boxed{\text{C}}$ Π $\boxed{\text{D}}$ w $\boxed{\text{E}}$ $\min$

Question 4
Attached

Dynamic Programming

Consider the *length of a longest common subsequence* (LLCS) of two sequences $\langle x_1, \ldots, x_m \rangle$ and $\langle y_1, \ldots, y_n \rangle$. For example, $X = \langle A, B, C, B, D, A, B \rangle$ and $\langle B, D, C, A, B, A \rangle$ have common subsequences of length 4, such as $\langle B, C, B, A \rangle$ and $\langle B, D, A, B \rangle$, but no longer ones. Define the ith *prefix* of a sequence $Z = \langle z_1, \ldots, z_\ell \rangle$ as $Z_i = \langle z_1, \ldots, z_i \rangle$, for $i \in 0 \mathinner{..} \ell$. In our example, X_3 is $\langle A, B, C \rangle$ and X_0 is the empty sequence. Consider the following recurrence — with placeholders α_1, α_2, β_1, β_2, and γ — on a numeric quantity $L[i, j]$:

$$L[i,j] = \begin{cases} 0 & \text{if } \beta_1 \\ L[\alpha_1, \alpha_2] + 1 & \text{if } \neg\beta_1 \text{ and } \beta_2 \\ \gamma\ \{L[\alpha_1, j],\ L[i, \alpha_2]\} & \text{otherwise} \end{cases}$$

Question 1: If $L[m, n]$ is returned by a correct algorithm for computing the LLCS of two sequences $X = \langle x_1, \ldots, x_m \rangle$ and $Y = \langle y_1, \ldots, y_n \rangle$, then what is $L[i, j]$, for $i \in 0 \mathinner{..} m$ and $j \in 0 \mathinner{..} n$?

- $\boxed{\text{A}}$ the LLCS of X and Y
- $\boxed{\text{B}}$ the LLCS of $\langle x_0, \ldots, x_i \rangle$ and $\langle y_0, \ldots, y_j \rangle$
- $\boxed{\text{C}}$ the LLCS of X_{i-1} and Y_{j-1}
- $\boxed{\text{D}}$ the LLCS of X_i and Y_j
- $\boxed{\text{E}}$ the LLCS of X_{i+1} and Y_{j+1}

Question 2: For the example above the recurrence, what is the sum $L[5, 3] + L[6, 4]$?

$\boxed{\text{A}}$ 3 $\boxed{\text{B}}$ 4 $\boxed{\text{C}}$ 5 $\boxed{\text{D}}$ 6 $\boxed{\text{E}}$ 7

Question 3: What is the Boolean placeholder β_1?

$\boxed{\text{A}}$ $i \cdot j = 0$ $\boxed{\text{B}}$ $i + j = 0$ $\boxed{\text{C}}$ $i = 0$ $\boxed{\text{D}}$ $j = 0$ $\boxed{\text{E}}$ $i \le j$

Question 4: What is the Boolean placeholder β_2?

$\boxed{\text{A}}$ $x_i \ne y_j$ $\boxed{\text{B}}$ $x_i < y_j$ $\boxed{\text{C}}$ $x_i \le y_j$ $\boxed{\text{D}}$ $x_i \ge y_j$ $\boxed{\text{E}}$ $x_i = y_j$

Question 5: What is the index placeholder α_1?

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

Question 6: What is the index placeholder α_2?

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

Question 7: What is the single-argument set-operator placeholder γ?

$\boxed{\text{A}}$ argmax $\boxed{\text{B}}$ average $\boxed{\text{C}}$ median $\boxed{\text{D}}$ set-size $\boxed{\text{E}}$ max

Question 8: Assuming $L[0, 0]$ is the upper-left corner of the table L, what is an ordering of filling L ***using a bottom-up method***, without referring to yet non-computed elements?

$\boxed{\text{A}}$ columns left-to-right,
bottom-up in the columns

$\boxed{\text{B}}$ columns left-to-right,
top-down in the columns

$\boxed{\text{C}}$ rows bottom-up, left-to-right in rows

$\boxed{\text{D}}$ rows bottom-up, right-to-left in rows

$\boxed{\text{E}}$ rows top-down, right-to-left in rows

Question 5
Attached

Dynamic Programming

Consider lectures $\ell_1, \ell_2, \ldots, \ell_n$ for which the same classroom is requested. Each lecture ℓ_i has a start time s_i and a finish time f_i, where $0 \leq s_i < f_i < \infty$. We wish to select a largest subset of lectures of which no two overlap in time. If selected, then ℓ_i happens during the half-open time interval $[s_i, f_i)$. For example, consider the following set of lectures:

i	1	2	3	4	5	6	7	8	9	10	11
s_i	1	3	0	5	3	5	6	8	8	2	12
f_i	4	5	6	7	9	9	10	11	12	14	16

The subset $\{\ell_3, \ell_9, \ell_{11}\}$ has non-overlapping lectures but $\{\ell_1, \ell_4, \ell_8, \ell_{11}\}$ is larger. The latter is a largest subset of non-overlapping lectures; another largest subset is $\{\ell_2, \ell_4, \ell_9, \ell_{11}\}$.
Assume the lectures are sorted by monotonically increasing finish time: $f_1 \leq f_2 \leq \cdots \leq f_n$. Let B_{ij} be the lectures that can happen between ℓ_i and ℓ_j; note that $B_{ij} \subseteq \{\ell_{i+1}, \ldots, \ell_{j-1}\}$. In the above example: $B_{2,9} = \{\ell_4\}$; $B_{9,2} = \varnothing = B_{3,8} = B_{8,3} = B_{11,4}$; $B_{4,11} = \{\ell_8, \ell_9\}$. Create two fictitious lectures ℓ_0 and ℓ_{n+1} with $f_0 = 0$ and $s_{n+1} = f_n$. Consider the following recurrence, parametrised by $\langle \alpha_1, \alpha_2, \alpha_3, \beta, \gamma \rangle$, on a numeric quantity $C[i, j]$:

$$C[i,j] = \begin{cases} 0 & \text{if and only if } \beta \\ \gamma\ \{C[\alpha_1, k] + \alpha_2 + C[k, \alpha_3] \mid \ell_k \in B_{ij}\} & \text{if and only if } \neg\beta \end{cases}$$

Question 1: If $C[0, n+1]$ is returned by a correct algorithm for computing the size of a largest subset of non-overlapping lectures, then what is $C[i, j]$, with $0 \leq i, j \leq n+1$?

- A the size of B_{ij}
- B the size of $\{\ell_i, \ell_{i+1}, \ldots, \ell_j\}$
- C the size of $\{\ell_i, \ell_{i+1}, \ldots, \ell_j\} \setminus \{\ell_0, \ell_{n+1}\}$
- D the size of a largest subset of non-overlapping lectures in B_{ij}
- E the size of a largest subset of non-overlapping lectures in $\{\ell_i, \ell_{i+1}, \ldots, \ell_j\}$

Question 2: For the example above, what is the sum $|B_{1,11}| + C[1, 11]$?

A 6   B 7   C 8   D 9   E other

Question 3: What is the Boolean condition β?

A $i = j$ B $i > j$ C $i \geq j$ D $B_{ij} = \varnothing$ E $B_{ij} \neq \varnothing$

Question 4: What is the index expression α_1?

A 1 B $i - 1$ C i D $j - 1$ E j

Question 5: What is the numeric expression α_2?

A -1 B $+1$ C size of B_{kk} D $C[k, k]$ E $f_k - s_k$

Question 6: What is the index expression α_3?

A $i - 1$ B i C $j - 1$ D j E n

Question 7: What is the single-argument set operator γ?

A argmax B average C max D median E set-size

Question 8: Assuming the desired quantity $C[0, n + 1]$ is in the upper-right corner of the table C, what is an ordering of filling C without referring to yet non-computed elements?

A columns left-to-right,
bottom-up in the columns

B columns left-to-right,
top-down in the columns

C rows top-down, left-to-right in rows

D rows top-down, right-to-left in rows

E rows bottom-up, right-to-left in rows

Question 6

Attached

Greedy Algorithms

Dynamic Programming

Consider lectures $\ell_1, \ell_2, \ldots, \ell_n$ for which the same classroom is requested. Each lecture ℓ_i has a start time s_i and a finish time f_i, where $0 \leq s_i < f_i < \infty$. We wish to select a largest subset of lectures of which no two overlap in time. If selected, then ℓ_i happens during the half-open time interval $[s_i, f_i)$. For example, consider the following set of lectures:

i	1	2	3	4	5	6	7	8	9	10	11
s_i	1	3	0	5	3	5	6	8	8	2	12
f_i	4	5	6	7	9	9	10	11	12	14	16

The subset $\{\ell_3, \ell_9, \ell_{11}\}$ has non-overlapping lectures but $\{\ell_1, \ell_4, \ell_8, \ell_{11}\}$ is larger. The latter is a largest subset of non-overlapping lectures; another largest subset is $\{\ell_2, \ell_4, \ell_9, \ell_{11}\}$.
Assume the lectures are sorted by monotonically increasing finish time: $f_1 \leq f_2 \leq \cdots \leq f_n$. Let B_{ij} be the lectures that can happen between ℓ_i and ℓ_j; note that $B_{ij} \subseteq \{\ell_{i+1}, \ldots, \ell_{j-1}\}$. In the example of Section : $B_{2,9} = \{\ell_4\}$; $B_{9,2} = \varnothing = B_{3,8} = B_{8,3} = B_{11,4}$; $B_{4,11} = \{\ell_8, \ell_9\}$. Create two fictitious lectures ℓ_0 and ℓ_{n+1} with $f_0 = 0$ and $s_{n+1} = f_n$. Consider the following recurrence, parameterised by $\langle \alpha_1, \alpha_2, \alpha_3, \beta, \gamma \rangle$, on a quantity $C[i, j]$:

$$C[i,j] = \begin{cases} 0 & \text{if } \neg\beta \\ \gamma\ \{C[\alpha_1, k] + \alpha_2 + C[k, \alpha_3] \mid \ell_k \in B_{ij}\} & \text{if } \beta \end{cases}$$

Question 1: If $C[0, n+1]$ is returned by a correct algorithm for computing the size of a largest subset of non-overlapping lectures, then what is $C[i, j]$, with $0 \leq i, j \leq n+1$?

- $\boxed{\text{A}}$ the size of B_{ij}
- $\boxed{\text{B}}$ the size of $\{\ell_i, \ell_{i+1}, \ldots, \ell_j\}$
- $\boxed{\text{C}}$ the size of $\{\ell_i, \ell_{i+1}, \ldots, \ell_j\} \setminus \{\ell_0, \ell_{n+1}\}$
- $\boxed{\text{D}}$ the size of a largest subset of non-overlapping lectures in B_{ij}
- $\boxed{\text{E}}$ the size of a largest subset of non-overlapping lectures in $\{\ell_i, \ell_{i+1}, \ldots, \ell_j\}$

Question 2: For the example of Section , what is the sum $|B_{2,11}| + C[0, 11]$?

$\boxed{\text{A}}$ 6 $\boxed{\text{B}}$ 7 $\boxed{\text{C}}$ 8 $\boxed{\text{D}}$ 9 $\boxed{\text{E}}$ other

Question 3: What is the Boolean condition β?

$\boxed{\text{A}}$ $B_{ij} = \varnothing$ $\boxed{\text{B}}$ $B_{ij} \neq \varnothing$ $\boxed{\text{C}}$ $i = 0$ $\boxed{\text{D}}$ $i > j$ $\boxed{\text{E}}$ $i \geq j$

Question 4: What is the index expression α_1?

$\boxed{\text{A}}$ $i - 1$ $\boxed{\text{B}}$ i $\boxed{\text{C}}$ $j - 1$ $\boxed{\text{D}}$ j $\boxed{\text{E}}$ k

Question 5: What is the numeric expression α_2?

$\boxed{\text{A}}$ 0 $\boxed{\text{B}}$ 1 $\boxed{\text{C}}$ $|B_{kk}|$ $\boxed{\text{D}}$ $C[k, k]$ $\boxed{\text{E}}$ $f_k - s_k$

Question 6: What is the index expression α_3?

$\boxed{\text{A}}$ $i - 1$ $\boxed{\text{B}}$ i $\boxed{\text{C}}$ $j - 1$ $\boxed{\text{D}}$ j $\boxed{\text{E}}$ k

Question 7: What is the single-argument set operator γ?

$\boxed{\text{A}}$ argmax $\boxed{\text{B}}$ argmin $\boxed{\text{C}}$ max $\boxed{\text{D}}$ min $\boxed{\text{E}}$ set-size

Question 8: Assuming the desired quantity $C[0, n + 1]$ is in the upper-right corner of the table C, what is an ordering of filling C without referring to yet non-computed elements?

$\boxed{\text{A}}$ rows top-down, left-to-right in rows

$\boxed{\text{B}}$ rows top-down, right-to-left in rows

$\boxed{\text{C}}$ rows bottom-up, right-to-left in rows

$\boxed{\text{D}}$ columns left-to-right,
top-down in the columns

$\boxed{\text{E}}$ columns left-to-right,
bottom-up in the columns

Question 7

Attached

Dynamic Programming

Consider lectures $\ell_1, \ell_2, \ldots, \ell_n$ for which the same classroom is requested. Each lecture ℓ_i has a start time s_i and a finish time f_i, where $0 \leq s_i < f_i < \infty$. We wish to select a largest subset of lectures of which no two overlap in time. If selected, then ℓ_i happens during the half-open time interval $[s_i, f_i)$. For example, consider the following set of lectures:

i	1	2	3	4	5	6	7	8	9	10	11
s_i	1	3	0	5	3	5	6	8	8	2	12
f_i	4	5	6	7	9	9	10	11	12	14	16

The subset $\{\ell_3, \ell_9, \ell_{11}\}$ has non-overlapping lectures but $\{\ell_1, \ell_4, \ell_8, \ell_{11}\}$ is larger. The latter is a largest subset of non-overlapping lectures; another largest subset is $\{\ell_2, \ell_4, \ell_9, \ell_{11}\}$.
Assume the lectures are sorted by monotonically increasing finish time: $f_1 \leq f_2 \leq \cdots \leq f_n$. Let B_{ij} be the lectures that can happen between ℓ_i and ℓ_j; note that $B_{ij} \subseteq \{\ell_{i+1}, \ldots, \ell_{j-1}\}$. In the example above: $B_{2,9} = \{\ell_4\}$; $B_{9,2} = \varnothing = B_{3,8} = B_{8,3} = B_{11,4}$; $B_{4,11} = \{\ell_8, \ell_9\}$. Create two fictitious lectures ℓ_0 and ℓ_{n+1} with $f_0 = 0$ and $s_{n+1} = f_n$. Consider the following recurrence, parameterised by $\langle \alpha_1, \alpha_2, \alpha_3, \beta, \gamma \rangle$, on a quantity $C[i, j]$:

$$C[i,j] = \begin{cases} 0 & \text{if } \beta \\ \gamma \ \{C[i, \alpha_1] + \alpha_2 + C[\alpha_3, j] \mid \ell_k \in B_{ij}\} & \text{if } \neg\beta \end{cases}$$

Question 1: If $C[0, n+1]$ is returned by a correct algorithm for computing the cardinality of a largest subset of non-overlapping lectures, then what is $C[i, j]$, with $0 \leq i, j \leq n+1$?

- A the cardinality of B_{ij}
- B the cardinality of a largest subset of non-overlapping lectures in B_{ij}
- C the cardinality of $\{\ell_i, \ell_{i+1}, \ldots, \ell_j\}$
- D the cardinality of a largest subset of non-overlapping lectures in $\{\ell_i, \ell_{i+1}, \ldots, \ell_j\}$
- E the cardinality of $\{\ell_i, \ell_{i+1}, \ldots, \ell_j\} \setminus \{\ell_0, \ell_{n+1}\}$

Question 2: For the example above, what is the sum $|B_{1,11}| + C[1, 11]$?

A 6   B 7   C 8   D 9   E other

Question 3: What is the Boolean condition β?

$\boxed{\text{A}}$ $B_{ij} = \varnothing$ $\boxed{\text{B}}$ $i = 0$ $\boxed{\text{C}}$ $j = n+1$ $\boxed{\text{D}}$ $i > j$ $\boxed{\text{E}}$ $i \geq j$

Question 4: What is the index expression α_1?

$\boxed{\text{A}}$ $j - 1$ $\boxed{\text{B}}$ $j + 1$ $\boxed{\text{C}}$ $k - 1$ $\boxed{\text{D}}$ k $\boxed{\text{E}}$ $k + 1$

Question 5: What is the numeric expression α_2?

$\boxed{\text{A}}$ 0 $\boxed{\text{B}}$ $|B_{kk}|$ $\boxed{\text{C}}$ $C[k, k]$ $\boxed{\text{D}}$ $f_k - s_k$ $\boxed{\text{E}}$ 1

Question 6: What is the index expression α_3?

$\boxed{\text{A}}$ $i - 1$ $\boxed{\text{B}}$ $i + 1$ $\boxed{\text{C}}$ $k - 1$ $\boxed{\text{D}}$ k $\boxed{\text{E}}$ $k + 1$

Question 7: What is the single-argument set operator γ?

$\boxed{\text{A}}$ set-size $\boxed{\text{B}}$ argmin $\boxed{\text{C}}$ argmax $\boxed{\text{D}}$ min $\boxed{\text{E}}$ max

Question 8: What is an ordering of the indices i and j under which the elements $C[i, j]$ of the table C can be filled bottom-up without referring to yet non-computed elements? (We say that $\langle a, b\rangle$ is *lexicographically smaller* than $\langle c, d\rangle$ if either $a < c$ or $a = c \wedge b < d$.)

$\boxed{\text{A}}$ by lexicographically increasing $\langle i, j\rangle$

$\boxed{\text{B}}$ by lexicographically decreasing $\langle i, -j\rangle$

$\boxed{\text{C}}$ by lexicographically decreasing $\langle i, j\rangle$

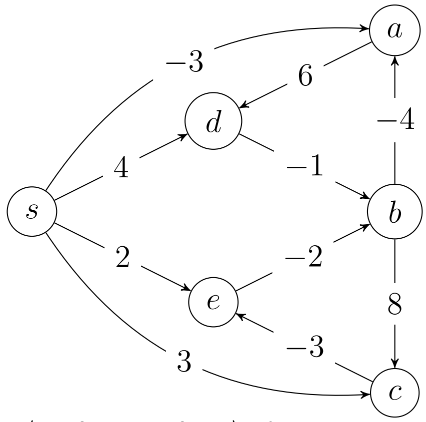
$\boxed{\text{D}}$ by lexicographically increasing $\langle j, i\rangle$

$\boxed{\text{E}}$ by lexicographically decreasing $\langle j, i\rangle$

Question 8
Attached

Dynamic Programming

Consider a weighted directed graph G with a set V of n vertices, a set E of m edges, and an edge-weight function $w\colon E \to \mathbb{R}$, with no negative-weight cycles, and a source vertex s, such as in the figure to the right.



Consider the following recurrence, parameterised by $\langle \alpha, \beta_1, \ldots, \beta_4, \gamma \rangle$, for a quantity $T[i, v]$:

$$T[i,v] = \begin{cases} 0 & \text{if } \alpha \wedge (v = s) \\ \beta_1 & \text{if } \alpha \wedge (v \neq s) \\ \gamma\left(T[i-1, \beta_2],\ \min\limits_{(x,v)\in E}(T[i-1,\beta_3] + \beta_4)\right) & \text{if } \neg\alpha \end{cases}$$

Question 1: If $T[n-1, v]$ is returned by a correct algorithm for computing the weight of a shortest path in G from s to v, then $T[i, v]$, with $0 \leq i < n$ and $v \in V$, denotes the weight of a shortest path from s to v with how many edges?

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

Question 2: For the graph above, what is the sum $T[3, a] + T[3, d]$?

$\boxed{\text{A}}$ -6 $\boxed{\text{B}}$ -4 $\boxed{\text{C}}$ -3 $\boxed{\text{D}}$ -2 $\boxed{\text{E}}$ -1

Question 3: What is the logical condition α?

$\boxed{\text{A}}$ $i = n-1$ $\boxed{\text{B}}$ $i = m-1$ $\boxed{\text{C}}$ $i = m$ $\boxed{\text{D}}$ $i = 0$ $\boxed{\text{E}}$ $i \cdot v = 0$

Question 4: What is the numeric expression β_1?

$\boxed{\text{A}}$ $+\infty$ $\boxed{\text{B}}$ $+1$ $\boxed{\text{C}}$ 0 $\boxed{\text{D}}$ -1 $\boxed{\text{E}}$ $-\infty$

Question 5: What is the index expression β_2?

$\boxed{\text{A}}$ $v-1$ $\boxed{\text{B}}$ v $\boxed{\text{C}}$ $v+1$ $\boxed{\text{D}}$ $m-1$ $\boxed{\text{E}}$ $n-1$

Question 6: What is the index expression β_3?

$\boxed{\text{A}}$ $v-1$ $\boxed{\text{B}}$ v $\boxed{\text{C}}$ $v+1$ $\boxed{\text{D}}$ x $\boxed{\text{E}}$ s

Question 7: What is the numeric expression β_4?

$\boxed{\text{A}}$ 1 $\boxed{\text{B}}$ $w(s,v)$ $\boxed{\text{C}}$ $w(s,x)$ $\boxed{\text{D}}$ $w(v,x)$ $\boxed{\text{E}}$ $w(x,v)$

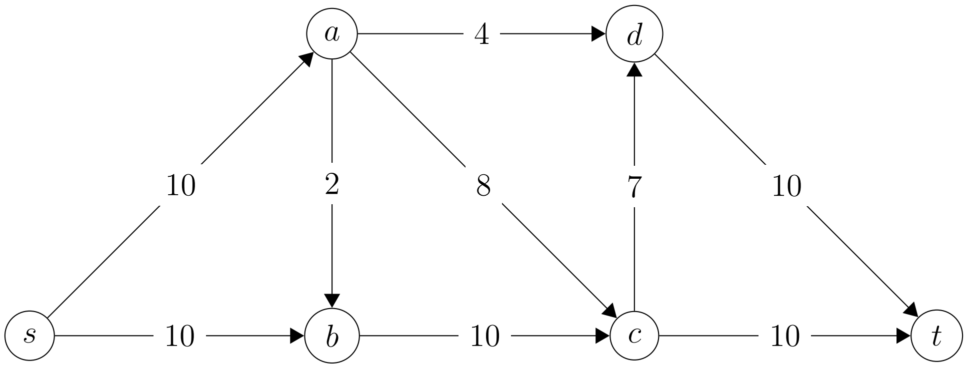
Question 8: What is the two-argument operator γ (written in prefix form above)?

$\boxed{\text{A}}$ $+$ $\boxed{\text{B}}$ Σ $\boxed{\text{C}}$ min $\boxed{\text{D}}$ Π $\boxed{\text{E}}$ w

Question 9
Attached

Maximum Flow

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



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

- $\boxed{\text{A}}$ $s \to b \to c \to a \to d \to t$, capacity +1
- $\boxed{\text{B}}$ none, the reached flow value is optimal
- $\boxed{\text{C}}$ $s \to b \to c \to d \to t$, capacity +1
- $\boxed{\text{D}}$ $s \to b \to a \to d \to t$, capacity +2
- $\boxed{\text{E}}$ $s \to b \to a \to c \to t$, capacity +2

Question 2: Are the flows across ***all*** cuts after the 3 augmentations of Question 1 equal?

$\boxed{\text{A}}$ yes: 16 $\boxed{\text{B}}$ yes: 17 $\boxed{\text{C}}$ yes: 18 $\boxed{\text{D}}$ yes: 19 $\boxed{\text{E}}$ no

Question 3: What is the maximum flow value (after ***all*** possible augmentations)?

$\boxed{\text{A}}$ 16 $\boxed{\text{B}}$ 17 $\boxed{\text{C}}$ 18 $\boxed{\text{D}}$ 19 $\boxed{\text{E}}$ 20

Question 4: What is the source set S of a minimum (s, t)-cut (S, T)?

- $\boxed{\text{A}}$ $\{s\}$
- $\boxed{\text{B}}$ $\{s, a\}$
- $\boxed{\text{C}}$ $\{s, b\}$
- $\boxed{\text{D}}$ $\{s, a, b\}$
- $\boxed{\text{E}}$ $\{s, a, b, c\}$