Iniziato	Friday, 23 June 2023, 14:33
Stato	Completato
Terminato	Friday, 23 June 2023, 14:48
Tempo impiegato	15 min.

Domanda **1**

Risposta corretta

Punteggio max.: 2,00

Let f, g be the functions defined as $f(n) = n2^{-15}n^2 \log n$ and $g(n) = \frac{2^{30}}{\log^{-1} n}n^3$.

Scegli una o più alternative:

- [x] $f \in \Theta(g)$ ✔
- [ ] $f \notin O(g)$
- [x] $f \in \Omega(g)$ ✔

Le risposte corrette sono: $f \in \Omega(g), f \in \Theta(g)$

Domanda **2**

Risposta corretta

Punteggio max.: 2,00

Turing Machines:

Scegli una o più alternative:

- [ ] Have finitely many configurations.
- [x] Can have as many tapes as needed. ✔
- [ ] Always halt their execution on any input.
- [x] Have finitely many states ✔

Le risposte corrette sono: Can have as many tapes as needed., Have finitely many states

Domanda **3**

Parzialmente corretta

Punteggio max.: 2,00

Which of the following assertions is true about the problem of checking whether two sequences of natural numbers contain exactly the same elements (both of them considered as part of the input)?

Scegli una o più alternative:

- [x] It is in $\mathbf{EXP}$ ✔
- [ ] It can be polynomially reduced to the 3SAT problem.
- [x] It is in $\mathbf{P}$ ✔
- [ ] It is well-known to be $\mathbf{NP}$-hard

Le risposte corrette sono: It is in $\mathbf{P}$, It is in $\mathbf{EXP}$, It can be polynomially reduced to the 3SAT problem.

Domanda **4**

Parzialmente corretta

Punteggio max.: 2,00

Which ones of the following inequalities between complexity classes are currently known to be true? Here, $\mathbf{NPC}$ stands for the class of all $\mathbf{NP}$-complete problems.

Scegli una o più alternative:

- [ ] $\mathbf{EXP} \neq \mathbf{NPC}$.
- [x] $\mathbf{P} \neq \mathbf{NPC}$. ✖
- [x] $\mathbf{EXP} \neq \mathbf{P}$. ✔
- [ ] $\mathbf{NP} \neq \mathbf{NPC}$.

La risposta corretta è: $\mathbf{EXP} \neq \mathbf{P}$.

Domanda **5**

Parzialmente corretta

Punteggio max.: 2,00

The concept class consisting of conjunctions of literals:

Scegli una o più alternative:

- ☐ Is not efficiently PAC-learnable, provided $\mathbf{RP} \neq \mathbf{NP}$.
- ☐ Has to do with rectangles.
- ☑ Is efficiently PAC-learnable. ✔
- ☐ Is easier to learn than the concept class of CNFs.

Le risposte corrette sono: Is efficiently PAC-learnable., Is easier to learn than the concept class of CNFs.

Vai a...

LAAI - 23062023 - Module 3 - Problems ▶

Domanda **1**

Completo

Punteggio max.: 6,00

Describe, formally or informally, a deterministic TM of the kind you prefer, which decides the following language:

$\mathcal{L} = \{w \in \{0,1\}^* \mid$ the number of occurrences of the symbol 0 in w is at most

Study the complexity of TM you have defined.

In the following TM we just use the input tape, we also use the states q_accept and q_reject to
to outputs the results of the computation.

(q_init, start) -> (q_count_0, start , R) // we start from the start symbol and move the tape on the right

(q_count_0, 0) -> (q_count_1, 0, R) // we encounter the 1st zero so we pass to the next state
(q_count_0, 1) -> (q_count_0, 1, R) // we read a one in input so we stay in the same state
(q_count_0, blank) -> (q_accept, blank, S) // we read blank from the input meaning that there are no symbol left to be read and we didn't count any zeros

(q_count_1, 0) -> (q_count_2, 0, R) // we encounter the 2nd zero so we pass to the next state
(q_count_1, 1) -> (q_count_1, 1, R) // we read a one in input so we stay in the same state
(q_count_1, blank) -> (q_accept, blank, S) // we read all the input with 1 occurences of zero so it belong to the language

(q_count_2, 0) -> (q_count_3, 0, R) // we encounter the 2rd zero so we pass to the next state
(q_count_2, 1) -> (q_count_2, 1, R) // we read a one in input so we stay in the same state
(q_count_2, blank) -> (q_accept, blank, S) // we read all the input with 2 occurences of zeros so it belong to the language

(q_count_3, 0) -> (q_count_4, 0, R) // we encounter the last zero admissible so we pass to the next state
(q_count_3, 1) -> (q_count_3, 1, R) // we read a one in input so we stay in the same state
(q_count_3, blank) -> (q_accept, blank, S) // we read all the input with 3 occurences of zeros so it belong to the language

(q_count_4, 0) -> (q_reject, 0, S) // we encountered our final zero meaning that the string doesn't belong to the language
(q_count_4, 1) -> (q_count_4, 1, R) // we read a one in input so we stay in the same state
(q_count_4, blank) -> (q_accept, blank, S) // we read all the input with 4 occurences of zerso so it belong to the language

The complexity of the TM is O(n) since it simply read the input from left to right and it halts as soon
as it founds a blank or 4 zeros.

Domanda **2**

Completo

Punteggio max.: 7,00

You are required to prove that the following problem $\mathcal{L}$ is in $\mathbf{NP}$. To do that, you can give a TM or define some pseudocode. The language $\mathcal{L}$ includes precisely those binary strings which are encodings of pairs in the form (G, k) where G is an undirected graph and k is a natural number such that there is a subset S of the set of edges of G having cardinality at most k and such that any node of G occurs in at least one of the edges in S.

In order to prove the problem to be in NP we have to find a:
- certificate polynomialy bounded in length
- verifier that works in polytime

The certificate is the subset S, since it is a subset of E its size is bounded and cannot be greater than |E| which is also cannot be greater than |V^2| so it's O(V^3) meaning that it's polynomially bounded in size.
The verifier basically check that S has size less than 4 and also that each node of V occurs at least one in the edges of S. It outputs 1 if the string belong to the language, 0 otherwise.

```
PSEUDOCODE ( G = (V, E) )
# checks the size of S
if len(S) > 4:    #O(|V^2|)
  return 0  # 1
else
 return 1 # 1

# checks that for each node in S they belong to at least to one edge in S
for node in V: #O(|V|)
 found = false
 for (u, v) in S: #O(|V^2|)
     if u == node || v == node
        found = true
 if found == false:
     return 0

return 1
```

in order to check if the verifier is polytime we have to look at:
- the number of instruction. which are polynomially bounded since it belongs to O(|V^3|)
- the size of the intermediate results, the only intermediate result is found which can only have 2 values. so the size of the intermediate results are polynomially bounded
- the execution time of each instruction, each instruction can be sinmply executed in polytime

In conclusion L belongs to P

Commento:
Why... 4?!?

Domanda **3**

Completo

Punteggio max.: 7,00

The degree of a node in an undirected graph is simply the number of edges which has this node as one of its endpoints. The degree of an undirected graph is the maximum degree of its nodes. Consider the problem $\mathtt{HAMPATHVAR} = \{G \mid G \text{ is a graph of degree 2 having an Hamiltonian path.}\}$. What's the complexity of $\mathtt{HAMPATHVAR}$.

```
G = (V, E)

# checks that each node has max 2 edges
for u in V:
 for

HAMPATHVAR belongs to P since we can find an polytime algorith that
solves the solution
```

Commento:
Your conclusion is correct, but there are no details about it.