Sia $S = \{a_1, a_2, \ldots, a_n\}$ un insieme di n attività che competono per l'uso di una stessa risorsa R, ove ciascuna attività è caratterizzata da un tempo di inizio utilizzo s_i e un tempo di fine utilizzo f_i, tali che $0 \leq s_i < f_i < \infty$, per $i = 1, 2, \ldots, n$.

Si supponga inoltre che la risorsa R risulti indisponibile a causa di manutenzione durante gli intervalli temporali $[x_j, y_j]$, con $0 \leq x_j < y_j$ e $j = 1, 2, \ldots, m$.

(a) Si descriva un algoritmo che selezioni il sottoinsieme di S che contiene il maggior numero possibile di attività mutuamente compatibili e compatibili anche con gli intervalli temporali in cui R non è disponibile.

(b) Si illustri l'algoritmo trovato al punto (a) nel caso in cui s_i, f_i, x_j, y_j siano dati dalle seguenti tabelle:

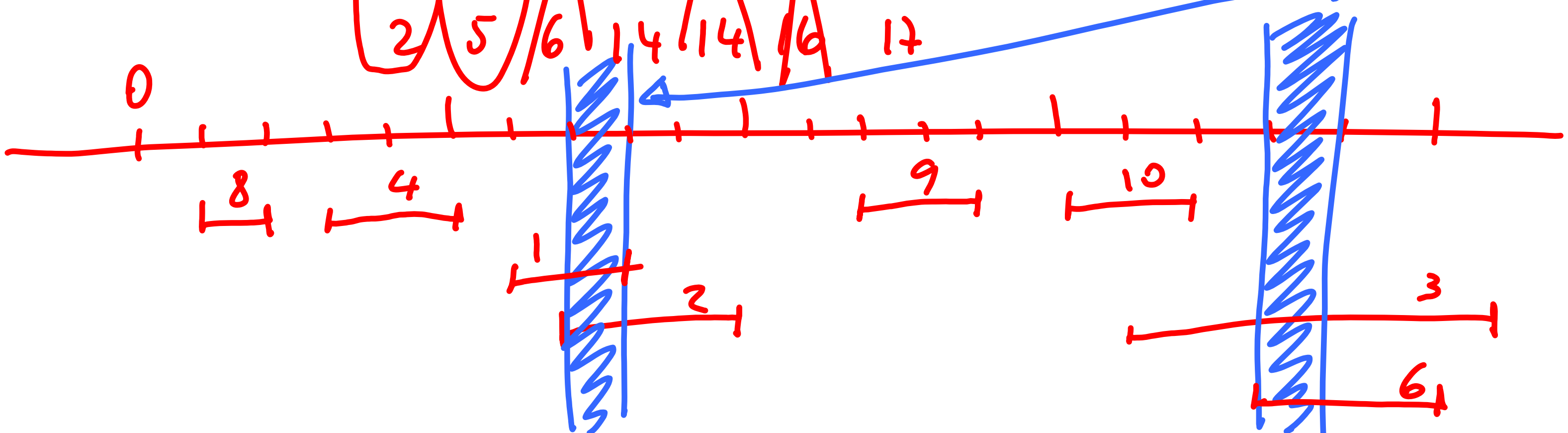
i	1	2	3	4	5	6	7	8	9	10	11
s_i	7	6	16	3	13	18	4	1	12	15	12
f_i	10	8	21	5	16	20	6	2	14	17	14

j	1	2
x_j	7	18
y_j	8	19

8	4	~~7~~	9	~~11~~	~~5~~	10
1	3	~~4~~	12	~~12~~	~~13~~	15
2	5	~~6~~	14	~~14~~	~~16~~	17

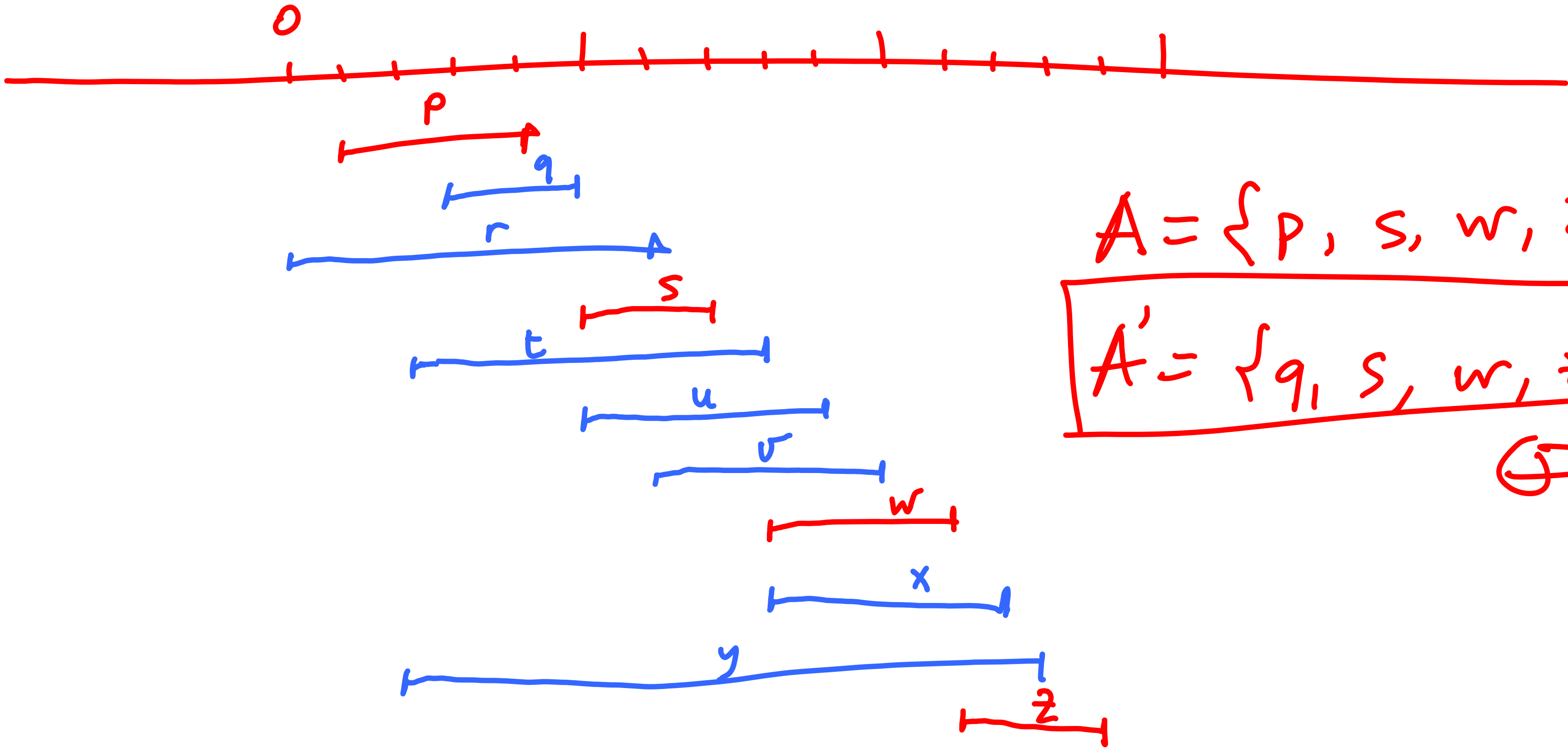
$A = \{8, 4, 9, 10\}$

vincoli



Nel contesto della metodologia *greedy*, si enunci con precisione il problema di ottimizzazione relativo alla *selezione di attività* e se ne discuta una soluzione efficiente, valutandone la complessità computazionale ed illustrandola sull'insieme di attività $S = \{p, q, r, s, t, u, v, w, x, y, z\}$ caratterizzate dai seguenti tempi iniziali e finali:

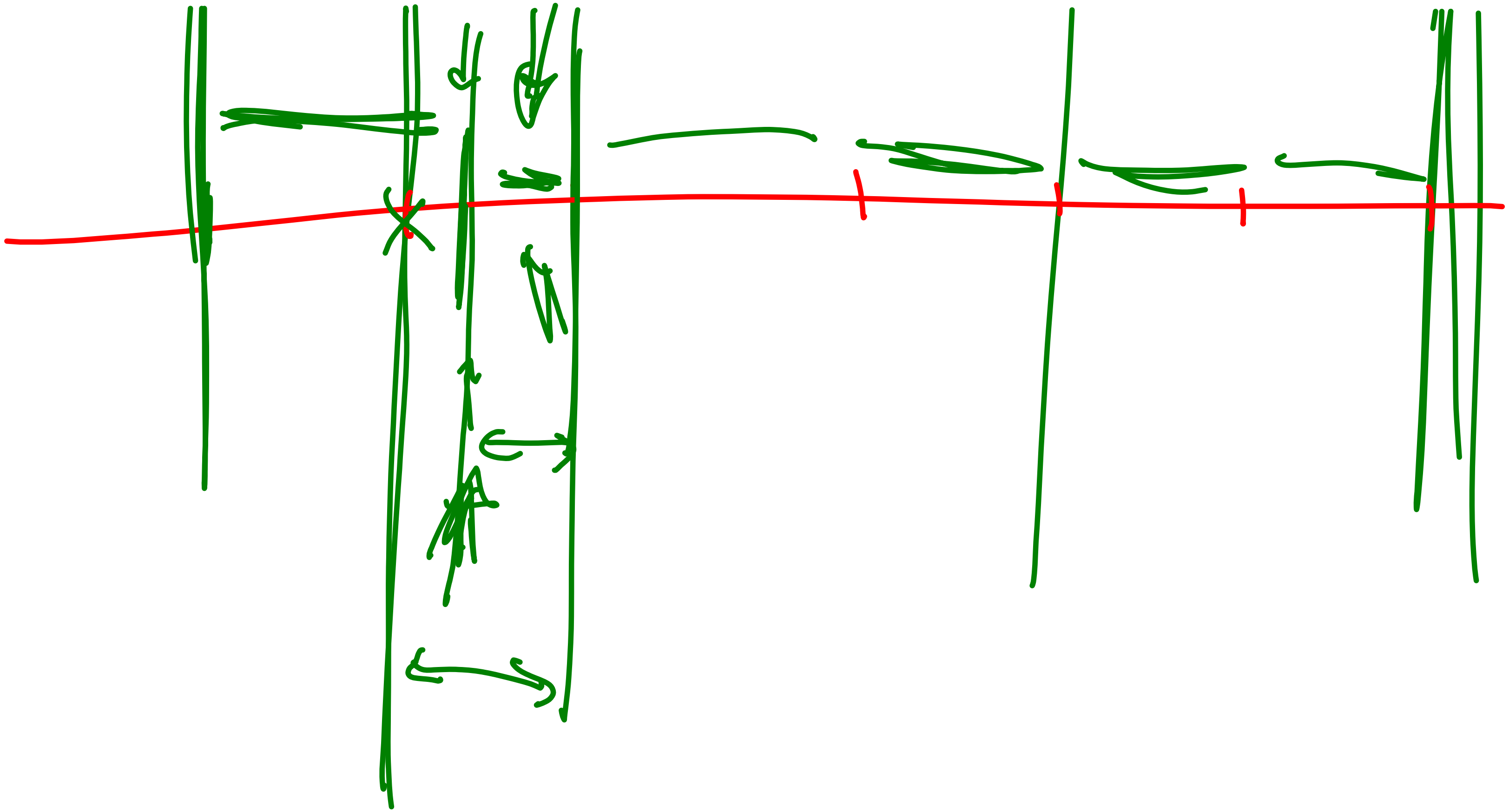
	SI	NO	NO	SI	NO	NO	NO	SI	NO	NO	SI
	p	q	r	s	t	u	v	w	x	y	z
inizio	1	3	0	5	3	5	6	8	8	2	12
fine	4	5	6	7	8	9	10	11	12	13	14



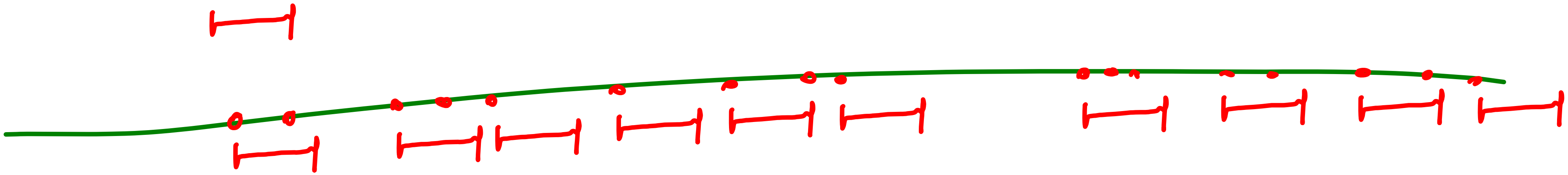
$A = \{p, s, w, z\}$

$\boxed{A' = \{q, s, w, z\}}$

Professor Midas drives an automobile from Newark to Reno along Interstate 80. His car's gas tank, when full, holds enough gas to travel n miles, and his map gives the distances between gas stations on his route. The professor wishes to make as few gas stops as possible along the way. Give an efficient method by which Professor Midas can determine at which gas stations he should stop, and prove that your strategy yields an optimal solution.



Describe an efficient algorithm that, given a set $\{x_1, x_2, \ldots, x_n\}$ of points on the real line, determines the smallest set of unit-length closed intervals that contains all of the given points. Argue that your algorithm is correct.



Si consideri il grafo orientato $\mathcal{G}$ rappresentato dalle seguenti liste di adiacenza:

A → B, H	D → E, G	G → D, H
B → C, D	E → F, G	H → D, F
C → A, F	F → D, G	

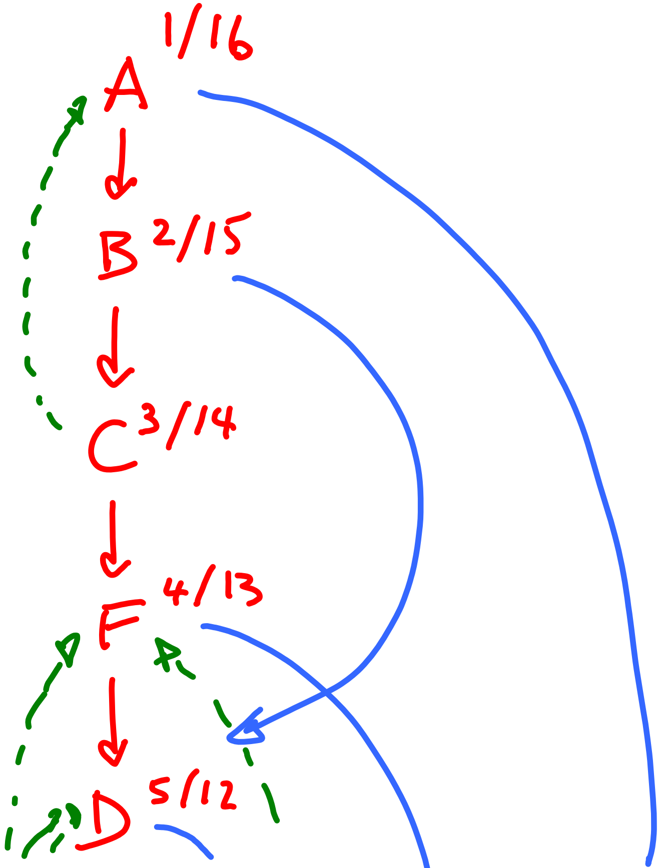
(A) Si descriva l'algoritmo di visita in profondità, fornendone anche lo pseudo-codice e determinandone la complessità computazionale. Quindi si effettui la visita in profondità del grafo $\mathcal{G}$ a partire dal vertice A (e poi procedendo lessicograficamente), indicando per ogni vertice i tempi di inizio e fine visita, e la classificazione di tutti gli archi.

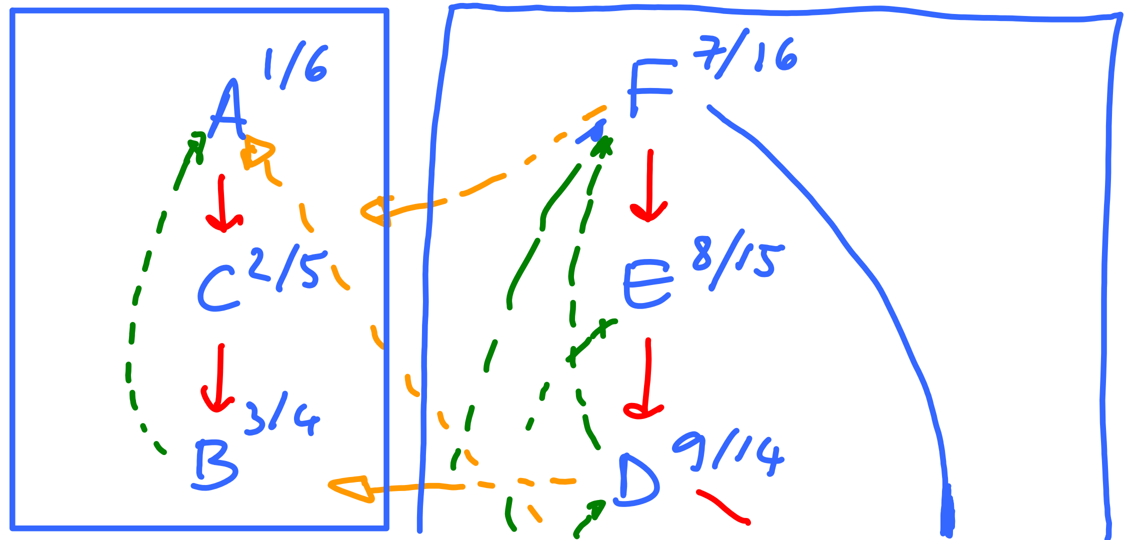
(B) Si definiscano le *componenti fortemente connesse* di un grafo orientato e si descriva un algoritmo per il loro calcolo.

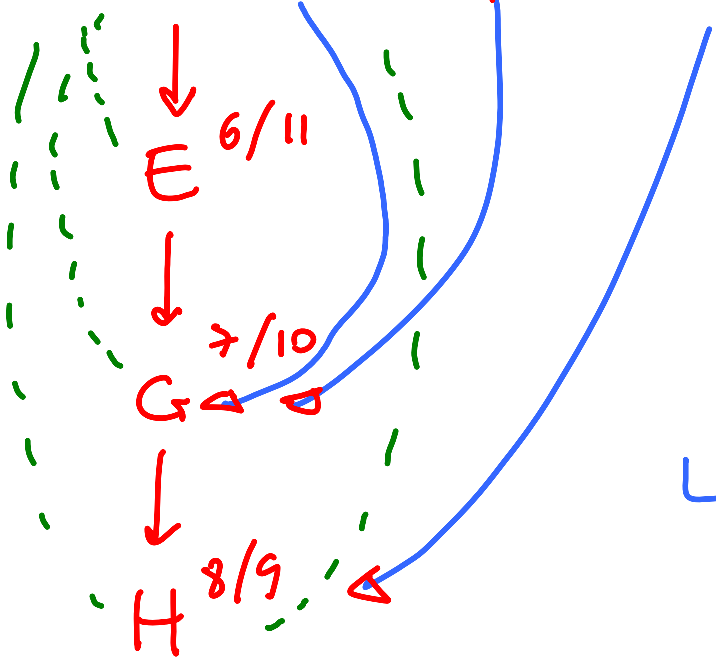
(C) Si determinino le componenti fortemente connesse del grafo $\mathcal{G}$.

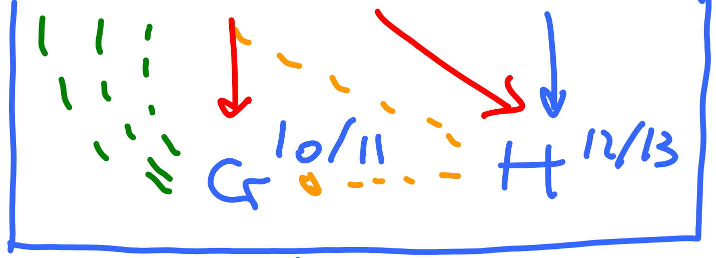
LISTE DI ADIACENZA DEL GRAFO TRASPOSTO G^T

A → ~~C~~	D → ~~B~~, ~~F~~, ~~G~~, ~~H~~	G → ~~D~~, ~~E~~, ~~F~~
B → ~~A~~	E → ~~D~~	H → ~~A~~, ~~G~~
C → ~~B~~	F → ~~C~~, ~~E~~, ~~H~~	









$$\underbrace{\{A, B, C\}, \quad \{D, E, F, G, H\}}_{\text{COMPONENTI FORTEMENTE CONNESSE}}$$

ESERCIZIO 2

Si consideri il seguente testo $\mathcal{T}$ di 59 caratteri

$$\mathsf{SI␣SENTIVANO␣CONTINUI␣CORI␣DI␣POPOPOPOPOPOPOPO␣POPOPOPOPOPOPOPO}$$

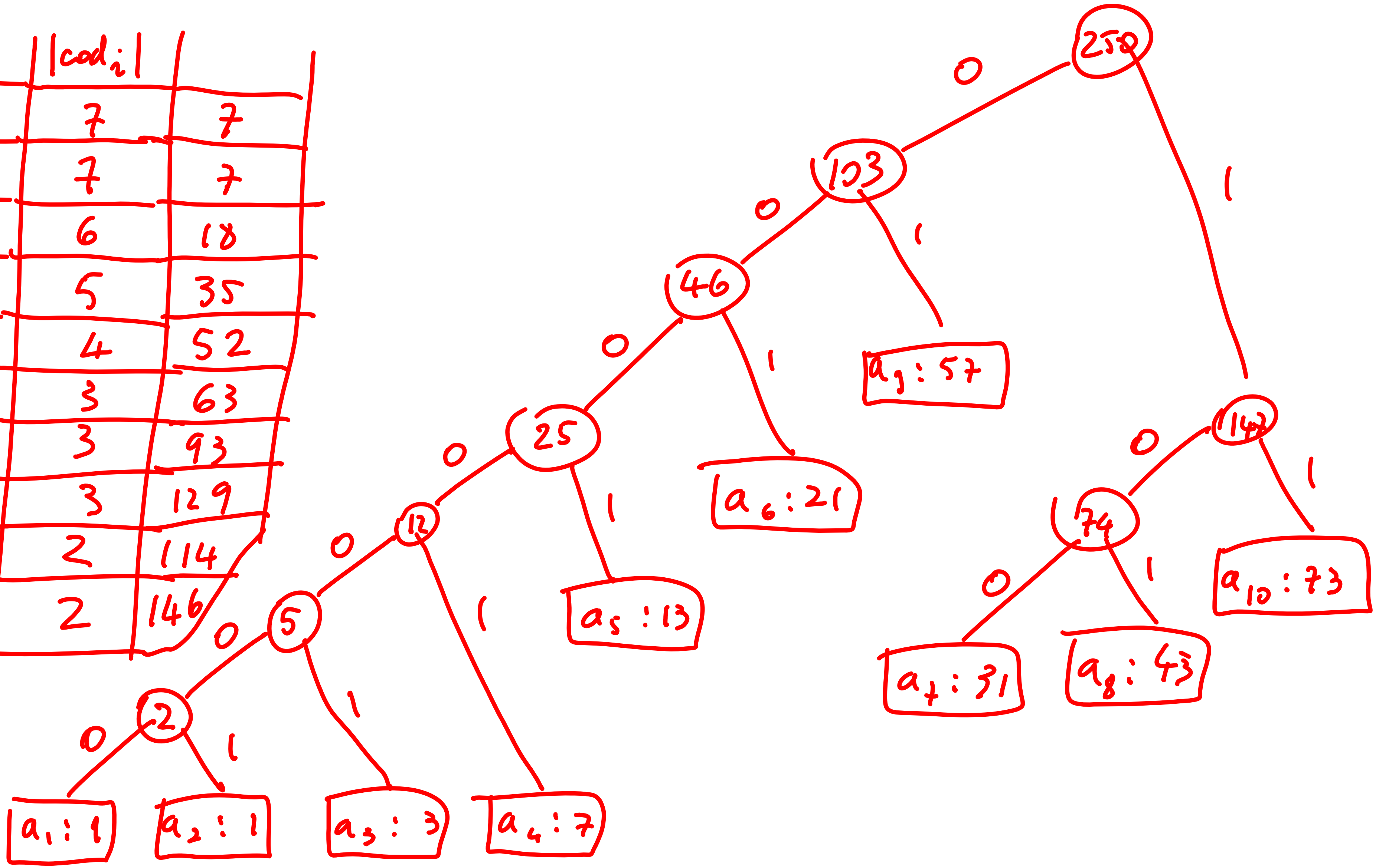
ove il simbolo "␣" rappresenta il blank.

Dopo avere illustrato l'algoritmo di Huffman, si trovi un codice prefisso binario per l'alfabeto dei simboli occorrenti in $\mathcal{T}$ che ne minimizzi la dimensione e si calcoli il risparmio in percentuale realizzato rispetto ad una rappresentazione di $\mathcal{T}$ mediante una codifica a lunghezza fissa minima.

Sia T un testo di 250 caratteri nell'alfabeto $\{a_1, \ldots, a_{10}\}$, ove la frequenza di a_i è data dall'espressione $f_i = i^2 - 3i + 3$, per $i = 1, \ldots, 10$.

Dopo aver dato la definizione di *codice prefisso*, si stabilisca qual è il numero minimo di bit necessari per rappresentare il testo T utilizzando un codice prefisso ottimo.

i	i^2	f_i	$\lvert cod_i \rvert$	
1	1	1	7	7
2	4	1	7	7
3	9	3	6	18
4	16	7	5	35
5	25	13	4	52
6	36	21	3	63
7	49	31	3	93
8	64	43	3	129
9	81	57	2	114
10	100	73	2	146



$$
\begin{array}{r}
7 + \\
7 + \\
18 + \\
35 + \\
52 + \\
63 + \\
93 + \\
129 + \\
114 + \\
146 \\
\hline
664
\end{array}
$$

← MINIMO NUMERO DI BIT PER LA CODIFICA DEL TESTO T CON UN CODICE (PREFISSO) A LUNGHEZZA VARIABILE

CODIFICHE:

a_1 : 0000000

a_2 : 0000001

a_3 : 000001

a_4 : 00001

a_5 : 0001

a_6 : 001

a_7 : 100

a_8 : 101

a_9 : 01

a_{10} : 11