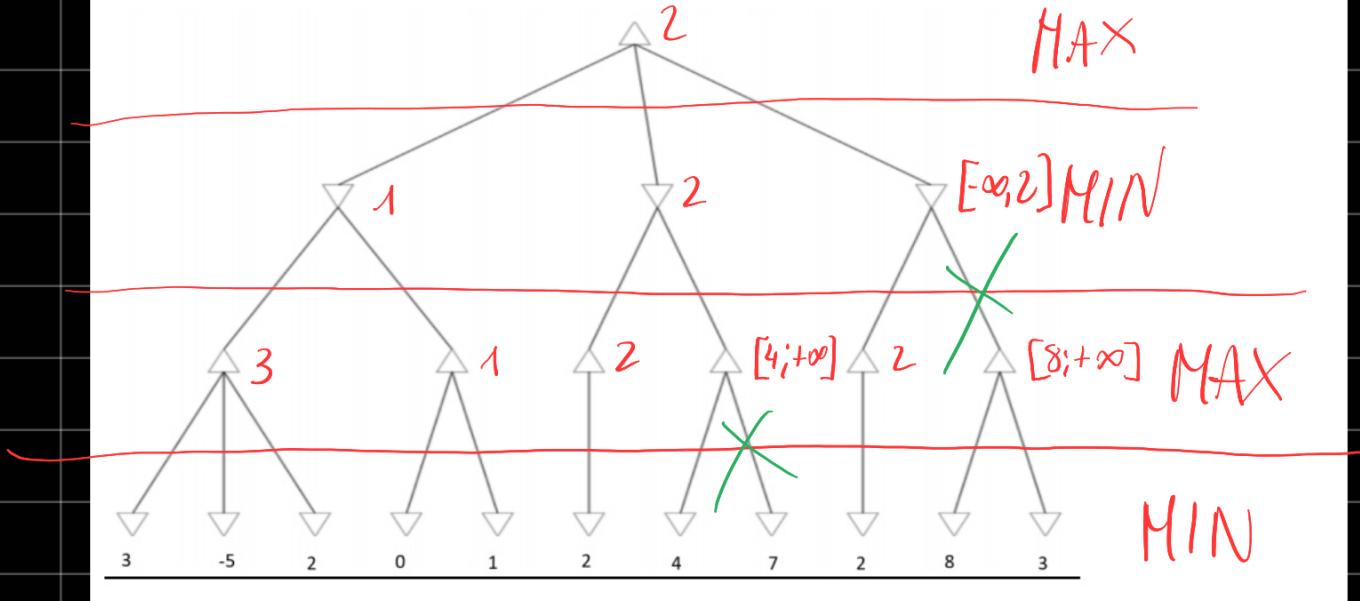


Esercizio 2

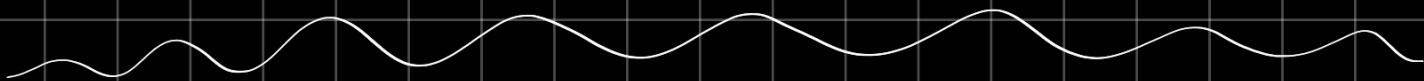
Si consideri l'albero di gioco riportato in basso, dove i punteggi sono dati dal punto di vista del primo giocatore.

- Supponendo che il primo giocatore sia MAX, si indichi quale mossa eseguirebbe e si spieghi il perché.
- La potatura alfa-beta su quali rami potrebbe agire?



a) eseguirebbe le seconda mossa, in quanto cerca il max tra i suoi figli $\{1, 2\}$, quindi: 2

b) vedi all'uso



Esercizio 1

Spiegare quando una funzione euristica $h(n)$ viene detta "ammissibile" e quando "consistente".

1) Un'euriistica è detta "ammissibile" quando assume un valore \leq di quello che è poi il valore reale del minimo.
Per poter dire che è ammissibile l'euriistica di un modo goal deve essere
E' detta invece consistente quando vale la diseguaglianza triangolare.

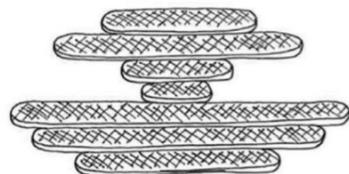
$$n$$

$$n \rightarrow n \leq n \rightarrow n \rightarrow n$$



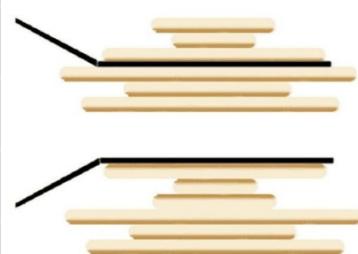
Esercizio 3

Si consideri un rompicapo ispirato al classico problema dei pancake: è necessario ordinare un certo numero di pancake (tutti di dimensioni differenti) impilati per dimensione, iniziando con quello più grande (in basso) e finendo con quello più piccolo (in alto). Di seguito un esempio.



L'obiettivo è fare il minor numero possibile di mosse con la spatola per ottenere la disposizione desiderata. Ogni mossa consiste nell'inserire la spatola ad una certa altezza

della pila, sollevarla con tutti i pancake che si trovano sopra e, con una rapida torsione del polso, posizionarli capovolti sulla sommità della pila.



Dare una descrizione del problema come problema di ricerca.

* Vedi problemi simili su "moving AI"
3)

STATO INIZIALE → uno qualiasi degli stati

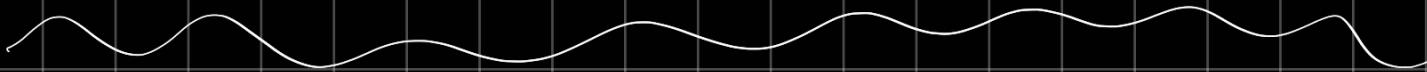
STATI → una qualsiasi delle combinaz. di pancake

AZIONI CONSENTITE → inserire la spatola sotto uno dei pancake

MODELLO DI TRANSIZ. → tutti i pancake sotto le spatole rimangono uguali, quelli sopra vengono capovolti in blocco

STATO FINALE → lo stato in cui ogni pancake si trova sopre ad un altro pancake più grande (eccetto chiaramente l'ultimo)

COSTI → 1 per ogni azione



Esercizio 4

a-1) Si consideri il seguente programma logico e se ne calcolino tutti gli answer set, **illustrando adeguatamente il procedimento seguito.**

```
time(0).    time(1).    time(2).  
energy(1).  
money(1,2).  
  
power(A,B) | energy(B) :- time(A), B=A+1, not time(B).  
power(A,B) :- energy(A), energy(B), not time(A).  
time(C) :- money(A,B), not power(B,A), C=B-1.  
energy(A) :- power(B,A), time(A), not time(A).
```

CONTRADDIZ.

4)

ISTANZIAZIONE

~~power(2,3) | energy(3) :- time(2), 3=2+1, not time(3).~~
~~power(3,3) :- energy(3), energy(3), not time(3).~~
~~power(3,1) :- energy(3), energy(1), not time(3).~~

~~time(1) :- Money(1,2), not power(2,1).~~

~~energy(3) :- power(2,3), time(3), not time(3).~~ ↳ sempre falsa

AS: {

{ [... EDB...], power(2,3), time(1) }

{ [... EDB...], energy(3), power(3,3), power(3,1), time(1) }

a-2) Si aggiunga il seguente strong constraint al programma del punto precedente. Come influisce sulle soluzioni del programma? Perché? Motivare adeguatamente la risposta.

```
:- power(X,Y), power(X,Z), #count{T : power(X,T)} > 1.
```

Q-2) questo strong regularity in seconds AS, in quanto
 abbiamo due power che metterebbero $\{\text{power}(3,1), \text{power}(3,3)\}$
 ed il count troverebbe $\{1,3\}$, quindi > 1

b) Si consideri ora un programma P (non è necessario sapere come è fatto) i cui answer set sono già stati calcolati e sono riportati di seguito.

```

A1: {set(1,2), set(2,3), top(0), top(1), top(2), top(4), gg(1), gg(3),
      gg(5), all(3,1), all(3,3), all(3,5), all(5,1), all(5,3), all(5,5)}
A2: {set(1,2), set(2,3), top(0), top(1), top(2), top(4), gg(1), gg(3),
      all(4,5), all(3,1), all(3,3)}
A3: {set(1,2), set(2,3), top(0), top(1), top(2), top(4), gg(1), all(2,3),
      gg(5), all(5,1), all(5,5)}
A4: {set(1,2), set(2,3), top(0), top(1), top(2), top(4), gg(1), all(2,3),
      all(4,5)}
  
```

Si supponga di aggiungere i seguenti weak constraint al programma P. Si calcoli quale sarebbe il costo di ognuno degli answer set riportati sopra: si riporti il **costo dettagliato per**

ciascun answer set e si indichi quello ottimo, commentando adeguatamente il procedimento seguito.

```

:~ all(X,Y), gg(X).           [1@2, X,Y]
:~ gg(X), top(Y), X>Y.       [1@X, X,Y]
  
```

b)

Costo A1: 4@5, 3@3, 6@2, 1@1

:~ all(3,1), gg(3)	[1@2, 3, 1]	6@2
:~ all(3,3), gg(3)	[1@2, 3, 3]	
:~ all(3,5), gg(3)	[1@2, 3, 5]	
:~ all(5,1), gg(5)	[1@2, 5, 1]	
:~ all(5,3), gg(5)	[1@2, 5, 3]	
:~ all(5,5), gg(5)	[1@2, 5, 5]	

gg(5), top(4), 5>4	[1@5, 5, 4]	4@5
gg(5), top(2), 5>2	[1@5, 5, 2]	
gg(5), top(1), 5>1	[1@5, 5, 1]	
gg(5), top(0), 5>0	[1@5, 5, 0]	
gg(3), top(2), 3>2	[1@3, 3, 2]	3@3
gg(3), top(1), 3>1	[1@3, 3, 1]	
gg(3), top(0), 3>0	[1@3, 3, 0]	

$gg(1), top(0), 1@0$ } $[1@1, 1, 0]$ } $1@1$

Costo A2: $3@3, 2@2, 1@1$

$all(3,1), gg(3)$ } $[1@2, 3, 1]$
 $all(3,3), gg(3)$ } $[1@2, 3, 3]$

$gg(3), top(2)$ } $[1@3, 3, 2]$
 $gg(3), top(1)$ } $[1@3, 3, 1]$
 $gg(3), top(0)$ } $[1@3, 3, 0]$

$gg(1), top(0)$ } $[1@1, 1, 0]$

Costo A3: $1@1, 2@2, 4@5$

Costo A4: $1@1$ OPTIMUM



Esercizio 5

La città di Pasticciopoli si è sempre distinta per la vivace vita pubblica. La città dispone di un sacco di strutture disponibili per la cittadinanza, e in particolare per le varie associazioni (dette "club") presenti sul territorio. La gestione delle strutture è demandata ad una istituzione molto famosa in città, il Coordinamento dei Club di Pasticciopoli. Il Coordinamento ha un presidente, e quello attuale ha terminato il proprio mandato: è quindi necessario eleggerne uno nuovo. Purtroppo, i vari club non si mettono d'accordo, e le votazioni si susseguono senza che ci sia un nome in grado di raccogliere abbastanza voti. La presidente uscente, Sergio Tammarella, conosce di fama il nostro amico Ciccio Pasticcio, e lo convoca per chiedergli aiuto: Ciccio deve organizzare una serie di gruppi di discussione in modo efficace, sì che da essi venga fuori un nome su cui tutti i club possano (in larga maggioranza, almeno) far convergere i propri voti. Tammarella concede a Ciccio una manciata di giorni e tutte le aule della sua residenza, il QuiSiVale: si scriva un programma ASP che organizzi questi gruppi di discussione tenendo conto delle specifiche riportate di seguito.

- Un gruppo di discussione è identificato da una coppia "aula-data". Tutti gli elettori possono in linea di principio partecipare a qualunque gruppo di discussione.
- [c1] Un elettore non può partecipare a più di un gruppo di discussione nello stesso giorno, a meno che non sia un coordinatore.
- [c2] In ogni caso, preso un qualunque intervallo di due giorni consecutivi, nessun elettore può partecipare a più di quattro gruppi di discussione in quell'intervallo, nemmeno i coordinatori.
- [c3] In un gruppo di discussione possono esserci al più 2 club non rappresentati (in altre parole, deve esserci almeno un elettore per ciascuno dei club, e sono ammesse al massimo due eccezioni per ogni gruppo di discussione).

- [w1] Si vuole minimizzare il numero di gruppi di discussione in cui accade che il club con più partecipanti a quel gruppo non è quello con il maggior numero di iscritti totali (tra i partecipanti al gruppo).
- [w2] Cosa più importante: si vogliono ridurre i casi di elettori impegnati in più gruppi rispetto ad elettori più giovani di loro

Modello dei dati in input:

data(D).	<- tutte le date in cui si tengono riunioni di gruppi di discussione
club(C, Membri).	<- i club, ciascuno col proprio numero di iscritti
coordinatore(E, Club).	<- i coordinatori dei club
gruppo(Aula, Data).	<- i gruppi di discussione
elettore(Nome, Club, Eta).	<- tutti gli elettori

- partecipa(N, A, D) | partecipa(N, A, D) :- elettore(N, -, -), gruppo(A, D).

% C1

nomeCoorol(N) :- coordinatore(N, -).

:- partecipa(N, A1, D), partecipa(N, A2, D), A1 != A2, not nomeCoorol(N).

% C2

:- elettore(N, -, -), data(D1), data(D2), D2 = D1 + 1,

$\hookrightarrow \#\{N, A, D1 : \text{partecipa}(N, A, D1)\} = P1, \#\{N, A, D2 : \text{partecipa}(N, A, D2)\} = P2, P1 + P2 > 4.$

% C3

clubTotali(T) :- #count{C : club(C, -)} = T.

:- gruppo(A, D), R = #count{C : partecipa(N, A, D), elettore(N, C, -)}, clubTotali(T), R < T - 2.

% W2

numPartecipazioni(E, N, Y) :- elettore(E, -, Y), N = #count{A, D : partecipa(E, A, D)}.

:- numPartecipazioni(E1, N1, Y1), numPartecipazioni(E2, N2, Y2), E1 != E2, >
 $\hookrightarrow Y1 > Y2, N1 > N2.$ [1@2, E1, E2]

% W1

partecipantiClub(C, N, A, D) :- club(C, -), gruppo(A, D), >

$\hookrightarrow N = \#\{E : \text{partecipa}(E, A, D), \text{elettore}(E, C, -)\}.$

`clubPiuNum(A,D,C) :- gruppo(A,D), M = #max{N,C : partecipanti(Club(C,N,A,D))},
partecipanti Club(C,M,A,D).`

`:~ clubPiuNum(A,D,C1), club(C2,M), M = #max{N,C : club(C,N)}, C1 != C2.
[1@1,C1,C2,A,D]`

