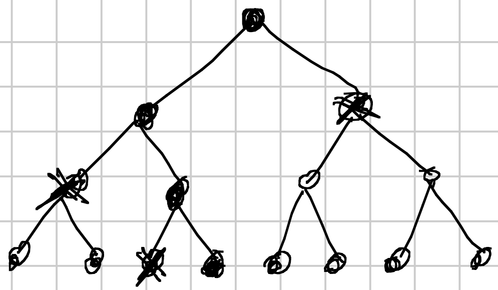
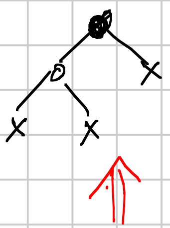
ALG. DOPPIO TORNEO



n

albero perfettamente bilanciato

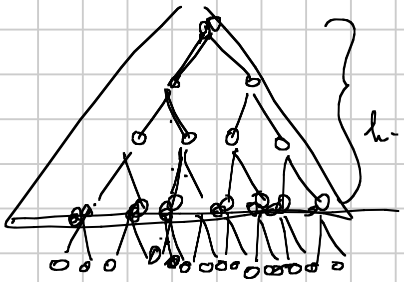
- n° partite = n° confronti = $n-1$ per stabilire il campione (max)

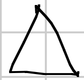


- stabilisce il vice campione o (secondo)

$C = n-1 + \log n - 1 = n + \log n - 2$ è anche un limite inferiore

$h = 0$

$h = 4$ h  $h-1$

albero perf. bilanciato 

altezza = n° archi da radice a foglie

Th: Un alb. perf. bilanciato di altezza h

h : $N = 2^{h+1} - 1$ nodi

$n = 2^h$ foglie

$i = 2^h - 1$ nodi interni

$n_h = 2 \cdot 2^{h-1} = 2^h$ vero

$i_h = 2^h - 1$ vero

Prova: induzione su h

base: $h = 0 \Rightarrow N = 2^1 - 1 = 1$ vero

$n = 2^0 = 1$ vero

$i = 2^0 - 1 = 0$ vero

passo induttivo:

$h-1 \Rightarrow h$ $N_h = N_{h-1} + 2 n_{h-1} =$

$= 2^h - 1 + 2 \cdot 2^{h-1} = 2^h - 1 + 2^h =$

$= 2^{h+1} - 1$ vero

n n° foglie $\qquad \underline{n} = 2^h \qquad \underline{h} = \log n$

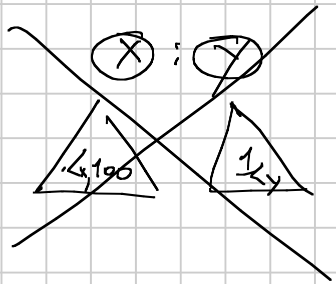
$C = n + \log n - 2$

$\underbrace{n-1}$ per stabilire il max (eventi contabili)

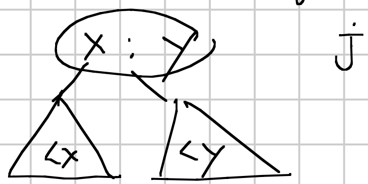
$$\begin{cases} M(j) = 1 & j = 1 \\ M(j) \leq 2M(j-1) + 1 \end{cases}$$

equazione di ricorrenza

numero di giocatori risultati minori del campione dopo j incontri



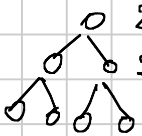
oracolo dice sempre il peggio



$$\begin{cases} M(j) = 1 & \text{se } j = 1 \\ M(j) \le 2\,M(j-1) + 1 \end{cases} \qquad M(j-1) \le 2M(j-2)+1$$

col metodo iterativo

$$M(j) \le 2\left(2M(j-2)+1\right)+1 \qquad M(j-2) = 2M(j-3)+1$$


2^0+
2^1+
2^2+
2^3-1

$$M(j) \le 2(2(2M(j-3)+1)+1)+1$$

$$M(j) \le 2(2(\ldots(2M(1)+1)+1)+1\ldots+1)+1$$

$$\le 2^{j-1}\cdot 1 + 2^{j-2} + 2^{j-3} + \ldots + 2 + 1 = \sum_{i=0}^{j-1} 2^i = \boxed{2^j - 1}$$

alla fine $M(j) \ge n-1$ $\qquad 2^j - \cancel{1} \ge n - \cancel{1} \qquad j \ge \log n$

$j \geq \log n$

j n° incontri minimo fatti dal campione per vincere

j il numero minimo di perdenti dal campione

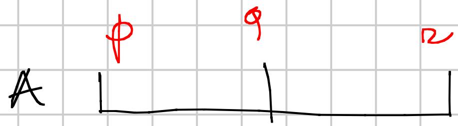
$$C \geq \underbrace{n-1}_{\text{max}} + \underbrace{\log n - 1}_{\text{vice max}} = n + \log n - 2$$

Algoritmo del doppio torneo $C = n + \log n - 2$

limite inferiore

ottimo

Determinare il max con alg. del torneo



TORNEO$(A, 1, n)$

TORNEO(A, p, r)

risolvi dir

```
if (p == r)        max = A[p];
   else {
```

dividi

```
   q = (p+r)/2;        Θ(1)
```

risolvi ricorsivamente

```
   < max1 >= TORNEO(A, p, q);
   < max2 >= TORNEO(A, q+1, r);
```

Combina

```
     if max1 > max2   max = max1;
            else   max = max2; }

return max;
```

$$\begin{cases} T(n) = \Theta(1) & n = 1 \\ T(n) = 2T\left(\frac{n}{2}\right) + \Theta(1) \end{cases}$$

$$T(n) = \Theta(n)$$

$$\boxed{C = n - 1}$$

Titolo nota 09/03/2016

EQUAZIONI DI RICORRENZA

$M(n) = 2M(n-1)+1$ metodo iterativo $2^n - 1$

$T(n) = 2T(\frac{n}{2}) + \Theta(1)$ Torneo $\Theta(n)$

$T(n) = 2T(\frac{n}{2}) + \Theta(n)$ Merge Sort $\Theta(n \log n)$

$T(n) = T(\frac{n}{2}) + \Theta(1)$ Ricerca Binaria $\Theta(\log n)$

albero di ricorsione

metodo iterativo

albero di ricorsione

metodo ~~di sostituzione~~ (Trial and error)

$\rightarrow$ metodo dell'esperto Teorema fondamentale ricorrenze

$$T(n) = a\,T\left(\frac{n}{b}\right) + f(n)$$

$a \geq 1$, $b > 1$

$f(n)$ definito sugli interi

1) se $f(n) = O\left(n^{\log_b a - \varepsilon}\right)$ per $\varepsilon > 0 \Rightarrow T(n) = \Theta\left(n^{\log_b a}\right)$

2) se $f(n) = \Theta\left(n^{\log_b a}\right) \Rightarrow T(n) = \Theta\left(n^{\log_b a} \cdot \log n\right)$

3) se $f(n) = \Omega\left(n^{\log_b a + \varepsilon}\right)$ per $\varepsilon > 0 \Rightarrow T(n) = \Theta(f(n))$

se $\underline{a\left(f\left(\frac{n}{b}\right)\right) \leq c\,f(n)}$ per $c < 1$

$$\boxed{T(n) = 9\,T\left(\frac{n}{3}\right) + n}$$

$a = 9$ $b = 3$ $f(n) = n$ $n^{\log_b a} =$

$= n^{\log_3 9} = n^2$

$\underline{n} : \underline{n^2}$ $\underline{n} = O\left(n^{2-\varepsilon}\right)$ $\varepsilon \leq 1$

Titolo nota 09/03/2016

caso 1 $\Rightarrow$ $T(n) = \Theta(n^{\log_b a}) = \Theta(n^2)$

$$T(n) = 2\,T\left(\frac{n}{2}\right) + \Theta(n)$$

$a = 2$, $b = 2$ $n^{\log_b a} = n$ $f(n) = n$

$f(n) = \Theta(n^{\log_b a})$? n

caso 2 : $T(n) = \Theta(n^{\log_b a} \cdot \log n) = \Theta(n \log n)$

$$T(n) = 2\,T\left(\frac{n}{2}\right) + \Theta(1)$$

$a = 2$, $b = 2$, $n^{\log_b a} = n$ $f(n) = \Theta(1) = n^0$?

n^0 ; $\underline{n}$ $f(n) = n^0 = O(n^{1-\varepsilon})$ $\forall\, \varepsilon \leq 1$

caso 1 $T(n) = \Theta(n^{\log_b a}) = \Theta(n)$

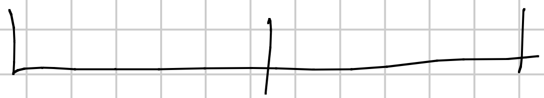
$$T(n) = T\left(\frac{2n}{3}\right) + \Theta(1)$$

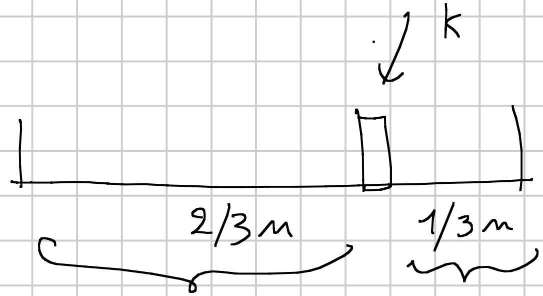
caso 2

$a = 1$ $b = \frac{3}{2}$ $n^{\log_{3/2} 1}$ n^0 : n^0 $\underline{\underline{T(n) = \Theta(\log n)}}$

$f(n) = \Theta(n^{\log_b a})$

Ricerca binaria





$$T(n) = T\left(\frac{2}{3}n\right) + \Theta(1)$$ caso pessimo