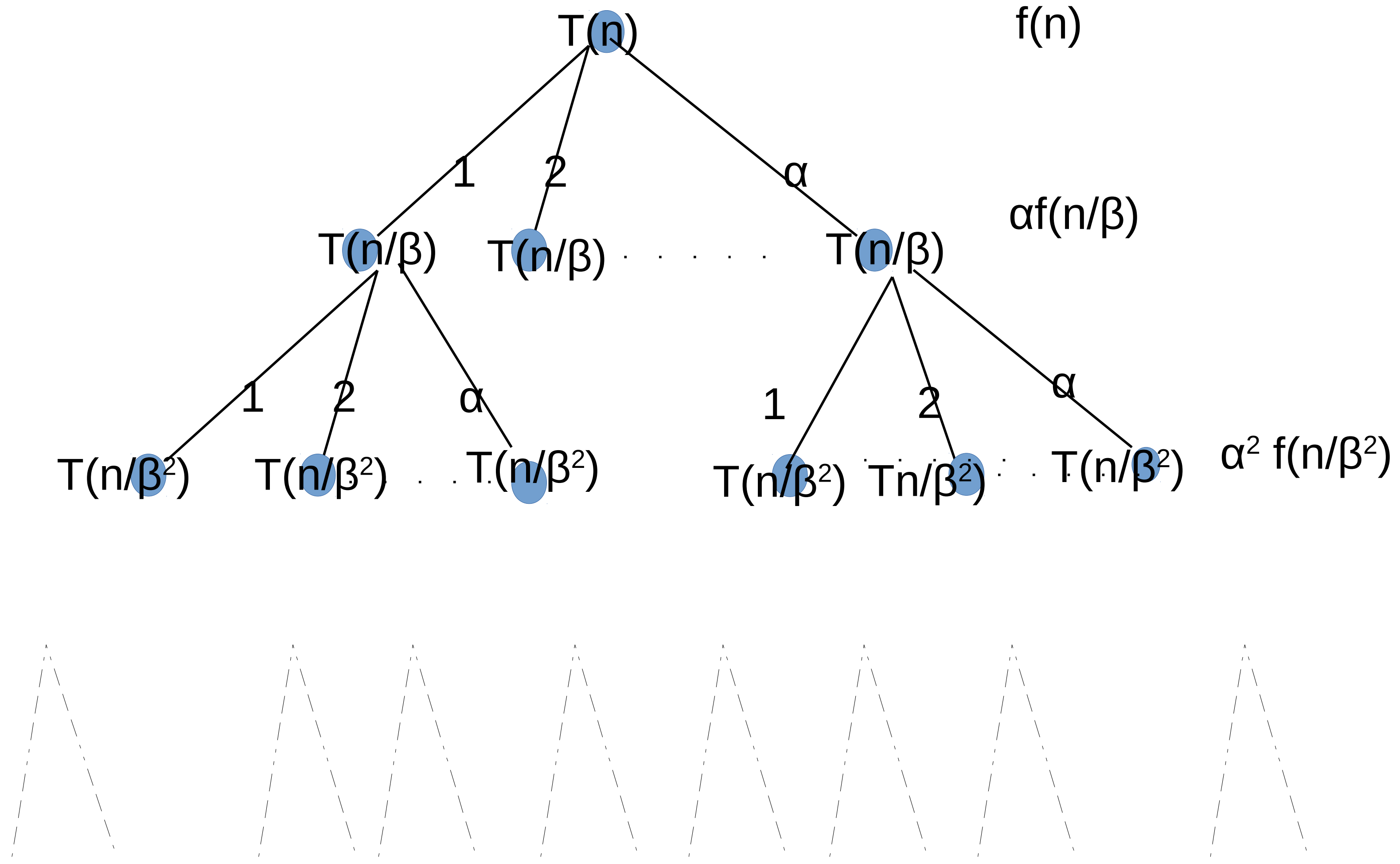
Soluzione delle relazioni di ricorrenza

- Consideriamo la seguente relazione di ricorrenza:

$$T(n) \leq \begin{cases} c_0 & \text{se } n \leq c \\ \alpha T(n/\beta) + f(n) & \text{altrimenti} \end{cases}$$

  con $\alpha \geq 1$ e $\beta > 1$ costanti.
- Nel caso in cui $T(n)$ sia la funzione che scaturisce dall'analisi di un algoritmo basato sul paradigma del Divide et Impera, $f(n)$ è il tempo per il lavoro di suddivisione e di ricombinazione. In altre parole, $f(n) = d(n) + r(n)$.
- In realtà nella ricorrenza n/β dovrebbe essere $\lceil n/\beta \rceil$ oppure $\lfloor n/\beta \rfloor$. Per stimare $T(n)$, assumiamo per semplicità che n sia una potenza di β in modo da poter omettere le parti intere superiori o inferiori.

Soluzione delle relazioni di ricorrenza



Sia h l’altezza dell’albero (h+1 livelli). Per i<h, Il tempo di esecuzione per tutte le chiamate ricorsive a livello i e` al piu` $\alpha^i f(n/\beta^i)$.

Soluzione delle relazioni di ricorrenza

- L'algoritmo non effettua chiamate ricorsive quando l'input ha dimensione al più c. Quindi la ricorrenza non sarà più applicata quando si arriva al livello i per cui per la prima volta $n/\beta^i \leq c$, cioè $i = \lceil \log_\beta n/c \rceil$.

- Il numero di livelli dell'albero è quindi $\lceil \log_\beta n/c \rceil + 1$ (partiamo dal livello 0) e ciascun nodo sul livello $\lceil \log_\beta n/c \rceil$ corrisponde al tempo $T(n/\beta^{\lceil \log_\beta n/c \rceil)}) \leq T(c) \leq c_0$. Il tempo totale per eseguire le $\alpha^{\lceil \log_\beta n/c \rceil}$ chiamate ricorsive in quest'ultimo livello è quindi $\leq \alpha^{\lceil \log_\beta n/c \rceil} c_0$.

- Abbiamo visto che per $i < \lceil \log_\beta n/c \rceil$, il tempo per eseguire tutte le chiamate sul livello i è $\alpha^i f(n/\beta^i)$.

- Sommando su tutti i livelli (compreso l'ultimo) si ha

$$T(n) \leq \alpha^{\lceil \log_\beta n/c \rceil} c_0 + \sum_{i=0}^{\lceil \log_\beta n/c \rceil - 1} \alpha^i f(n/\beta^i).$$

Soluzione delle relazioni di ricorrenza

- Vogliamo stimare la funzione

$$T(n) \leq \begin{cases} c_0 & \text{se } n \leq c \\ \alpha T(n/\beta) + f(n) & \text{altrimenti} \end{cases}$$

  quando la funzione $f(n)$ è limitata da $c'n^k$, dove c' e k sono due costanti tali che $k \geq 0$, $c' > 0$ ($f(n)$ polinomiale).
- Da quanto ottenuto nella slide precedente si ha che

$$\begin{aligned} T(n) &\leq \alpha^{\lceil \log_\beta n/c \rceil} c_0 + \sum_{i=0}^{\lceil \log_\beta n/c \rceil - 1} \alpha^i f(n/\beta^i) \\ &\leq \alpha^{\lceil \log_\beta n/c \rceil} c_0 + \sum_{i=0}^{\lceil \log_\beta n/c \rceil - 1} \alpha^i c'(n/\beta^i)^k \end{aligned}$$

Soluzione delle relazioni di ricorrenza

- Abbiamo visto che se $f(n) \leq c' n^k$, dove c' e k sono due costanti tali che $k \geq 0$, $c' > 0$, allora

$$T(n) \leq \alpha^{\lceil \log_\beta n/c \rceil} c_0 + c' n^k \sum_{i=0}^{\lceil \log_\beta n/c \rceil - 1} (\alpha/\beta^k)^i.$$

- consideriamo i 2 seguenti casi:
- $\alpha = \beta^k$: In questo caso si ha

$$\alpha^{\lceil \log_\beta n/c \rceil} c_0 = (\beta^k)^{\lceil \log_\beta n/c \rceil} c_0 < (\beta^k)^{\log_\beta (n/c)+1} c_0 = \beta^k (n/c)^k c_0 = O(n^k)$$

e

$$c' n^k \sum_{i=0}^{\lceil \log_\beta n/c \rceil - 1} (\alpha/\beta^k)^i = c' n^k \sum_{i=0}^{\lceil \log_\beta n/c \rceil - 1} 1 = c' n^k \lceil \log_\beta n/c \rceil = O(n^k \log_\beta n).$$

Quindi $T(n) = O(n^k) + O(n^k \log_\beta n) = O(n^k \log_\beta n) = O(n^k \log n)$.

- $\alpha \neq \beta^k$: In questo caso si ha

$$\begin{aligned}
\alpha^{\lceil \log_\beta n/c \rceil} c_0 &< c_0 \alpha^{\log_\beta n/c + 1} = c_0 \alpha \cdot \alpha^{\log_\beta n/c} = c_0 \alpha \cdot \alpha^{\log_\alpha (n/c) \log_\beta \alpha} \\
&= c_0 \alpha (n/c)^{\log_\beta \alpha} = O(n^{\log_\beta \alpha}), \qquad (1)
\end{aligned}$$

e

$$c' n^k \sum_{i=0}^{\lceil \log_\beta n/c \rceil - 1} (\alpha/\beta^k)^i = c' n^k \cdot \frac{(\alpha/\beta^k)^{\lceil \log_\beta n/c \rceil} - 1}{(\alpha/\beta^k) - 1}. \qquad (2)$$

Consideriamo i due sottocasi di $\alpha \neq \beta^k$: $\alpha < \beta^k$ e $\alpha > \beta^k$

- Caso $\alpha < \beta^k$:

$$\begin{aligned}\frac{(\alpha/\beta^k)^{\lceil \log_\beta (n/c) \rceil} - 1}{(\alpha/\beta^k) - 1} &= \frac{1 - (\alpha/\beta^k)^{\lceil \log_\beta n/c \rceil}}{1 - (\alpha/\beta^k)} \\ &< \frac{1}{1 - (\alpha/\beta^k)} = \frac{\beta^k}{\beta^k - \alpha} = O(1).\end{aligned}$$

Si ha quindi che la suddetta relazione insieme alla (1) e alla (2) della slide precedente implicano:

$$T(n) \leq O(n^{\log_\beta \alpha}) + c' n^k O(1) = O(n^{\log_\beta \alpha} + n^k).$$

Si noti che $\alpha < \beta^k$ implica $\log_\beta \alpha < k$ e di conseguenza si ha

$$T(n) = O(n^{\log_\beta \alpha} + n^k) = O(n^k).$$

Soluzione delle relazioni di ricorrenza

- Caso $\alpha > \beta^k$:

$$\frac{(\alpha/\beta^k)^{\lceil \log_\beta(n/c) \rceil} - 1}{(\alpha/\beta^k) - 1} < \frac{(\alpha/\beta^k)^{\log_\beta(n/c)+1} - 1}{(\alpha/\beta^k) - 1} = \frac{(\alpha/\beta^k)(\alpha/\beta^k)^{\log_\beta(n/c)} - 1}{(\alpha/\beta^k) - 1}$$

$$= O((\alpha/\beta^k)^{\log_\beta(n/c)}) = O((\alpha/\beta^k)^{\log_{(\alpha/\beta^k)}(n/c)\log_\beta(\alpha/\beta^k)})$$

$$= O((n/c)^{\log_\beta(\alpha/\beta^k)}) = O((n/c)^{\log_\beta \alpha - \log_\beta \beta^k})$$

$$= O(n^{\log_\beta(\alpha)-k})$$

Si ha quindi che la suddetta relazione insieme alla (1) e alla (2) della slide precedente implicano:

$$T(n) \leq O(n^{\log_\beta \alpha}) + c' n^k O(n^{\log_\beta(\alpha)-k}) = O(n^{\log_\beta \alpha} + n^k n^{\log_\beta(\alpha)-k}) = O(n^{\log_\beta \alpha}).$$

Soluzione delle relazioni di ricorrenza

- Abbiamo stimato la funzione

$$T(n) \leq \begin{cases} c_0 & \text{se } n \leq c \\ \alpha T(n/\beta) + c' n^k & \text{altrimenti,} \end{cases}$$

  dove c' e k sono due costanti tali che $k \geq 0$, $c' > 0$.

- Abbiamo provato

$$T(n) = \begin{cases} O(n^k) & \text{se } \alpha < \beta^k \\ O(n^k \log n) & \text{se } \alpha = \beta^k \\ O(n^{\log_\beta \alpha}) & \text{se } \alpha > \beta^k \end{cases}$$

- **Esempi:** Nel caso di MergeSort $\alpha = 2$, $\beta = 2$ e $k = 1$. Si ha $\alpha = \beta^k$ e quindi $T(n) = O(n^k \log n) = O(n \log n)$.
  Nel caso dell'algoritmo per la ricerca binaria $\alpha = 1$, $\beta = 2$ e $k = 0$. Si ha $\alpha = \beta^k$ e quindi $T(n) = O(n^k \log n) = O(\log n)$.

SOLUZIONE DELLE RELAZIONI DI RICORRENZA QUANDO n NON È POTENZA DI β

- Consideriamo la seguente relazione di ricorrenza:

$$T(n) = \begin{cases} c_0 & \text{se } n \leq c \\ \alpha T(n/\beta) + c' n^k & \text{altrimenti} \end{cases}$$

con $\alpha \geq 1$ e $\beta > 1$ costanti.

- Quando n non è una potenza di β la taglia di ciascun sottoproblema è $\lceil n/\beta \rceil$ oppure $\lfloor n/\beta \rfloor$.
- Siccome vogliamo stabilire un limite superiore per $T(n)$ mettiamoci nel caso peggiore in cui la taglia di ciascun sottoproblema è $\lceil n/\beta \rceil$ Consideriamo quindi la seguente relazione di ricorrenza:

$$T(n) = \begin{cases} c_0 & \text{se } n \leq c \\ \alpha T(\lceil n/\beta \rceil) + c' n^k & \text{altrimenti} \end{cases}$$

con $\alpha \geq 1$, $\beta > 1$ e $k \geq 0$ costanti.

- Usando questa nuova relazione di ricorrenza potremmo usare un procedimento simile a quello usato per il caso in cui n è potenza di β per provare le stesse limitazioni superiori viste per quel caso. Nel seguito invece useremo un argomento molto semplice per dedurre che quelle limitazioni valgono anche quando n non è potenza di β.

Soluzione delle relazioni di ricorrenza quando n non è potenza di β

- Sia p il più piccolo intero positivo per cui $n \leq \beta^p$, cioè p è l'intero per cui $\beta^{p-1} < n \leq \beta^p$.
- Osserviamo che siccome $T(n)$ è una funzione non decrescente allora $T(n) \leq T(\beta^p)$.
- Applicando a $T(\beta^p)$ la limitazione asintotica dimostrata per le potenze di β si ha

$$T(\beta^p) = \begin{cases} O((\beta^p)^k) & \text{se } \alpha < \beta^k \\ O((\beta^p)^k \log(\beta^p)) & \text{se } \alpha = \beta^k \\ O((\beta^p)^{\log_\beta \alpha}) & \text{se } \alpha > \beta^k \end{cases} \tag{3}$$

- Osserviamo che $\beta^p = \beta\beta^{p-1} < \beta n$, dal momento che $\beta^{p-1} < n$. Si ha quindi che

$$O((\beta^p)^k) = O((\beta n)^k) = O(\beta^k n^k) = O(n^k),$$

$$O((\beta^p)^k \log(\beta^p)) = O((\beta n)^k) \log(\beta n)) = O(n^k(\log(\beta)+\log n)) = O(n^k \log n),$$

$$O((\beta^p)^{\log_\beta \alpha}) = O((\beta n)^{\log_\beta \alpha}) = O(\beta^{\log_\beta \alpha} n^{\log_\beta \alpha}) = O(n^{\log_\beta \alpha}).$$

  La (3) può essere quindi scritta come segue

$$T(\beta^p) = \begin{cases} O(n^k) & \text{se } \alpha < \beta^k \\ O(n^k \log n) & \text{se } \alpha = \beta^k \\ O(n^{\log_\beta \alpha}) & \text{se } \alpha > \beta^k \end{cases}$$

- Poichè $T(n) \leq T(\beta^p)$ allora le limitazioni appena provate valgono anche per $T(n)$.

Esempi di relazioni di ricorrenza della forma $T(n) \leq \alpha T(n/\beta) + cn^k$

- Ricerca binaria

$$T(n) \leq \begin{cases} c_0 & \text{se } n \leq 1 \text{ oppure } k \text{ è l'elemento centrale} \\ T(n/2) + c & \text{altrimenti} \end{cases}$$

Si ha $\alpha = 1, \beta = 2,\ k = 0$.

Siccome $\alpha = \beta^k$, siamo nel secondo caso e si ha

$$T(n) = O(n^k \log n)) = O(\log n).$$

Esempi di relazioni di ricorrenza della forma $T(n) \leq \alpha T(n/\beta) + n^k$

Nell'ordinamento per fusione,

$$T(n) \leq \begin{cases} c_0 & \text{se } n \leq 1 \\ 2T(n/2) + cn & \text{altrimenti} \end{cases}$$

Quindi,

- $\alpha = 2$, $\beta = 2$ e $k = 1$
- siamo nel caso $\alpha = \beta^k$ e quindi $T(n) = O(n^k \log n) = O(n \log n)$.

Esempi di relazioni di ricorrenza della forma $T(n) \leq \alpha T(n/\beta) + n^k$

$$T(n) \leq \begin{cases} c_0 & \text{se } n \leq 1 \\ T(n/2) + cn & \text{altrimenti} \end{cases}$$

Una relazione di questo tipo è quella che scaturisce dall'analisi per il caso ottimo' di QuickSelect. Qui tralasciamo il caso in cui l'elemento da selezionare è proprio il pivot.
Quindi,

- $\alpha = 1$, $\beta = 2$ e $k = 1$
- siamo nel caso $\alpha < \beta^k$ e quindi $T(n) = O(n^k) = O(n)$.