Lezione del 25 Marzo 2020 - Appunti

Complessità del Merge sort iterativo

```
  // Credits: Prof. Shun Yan Cheung - Emory College
  public static void sort(double[] a)
 {
    int width;
    for ( width = 1; width < a.length; width = 2*width )
    {
       // Combine pairs of array a of width "width"
       int i;
       for ( i = 0; i < a.length; i = i + 2*width )
       {
          int left, middle, right;
          left = i;
          middle = i + width;
          right  = i + 2*width;
          merge( a, left, middle, right );
              // Merge è quello solito
       }
    }
 }
```

Supponiamo che la variabile k conti il numero dei cicli più esterni (quelli in cui aggiorniamo `width`). Poniamo a 0 il valore iniziale di k.

Per ogni valore di k, calcoliamo il costo di caso peggiore del ciclo ``for` più interno.

- $k = 0$. Abbiamo n array di dimensione 1 ed effettuiamo $n/2$ `merge`. Ogni merge ha costo al più $2c$, dove c è una costante abbastanza grande. Il costo del primo ciclo è quindi al più $(c + 2c) \cdot (n/2)$.
- $k = 1$. Abbiamo $n/2$ array di dimensione $2 = 2^1$ ed effettuiamo $n/4 = n/2^2$ merge. Il costo di ciascun merge è al più $4c$. Il costo complessivo del secondo ciclo è al più pari a $(c + 4c) \cdot (n/2^2)$.
- ....
- Generico valore di k: Abbiamo $n/2^k$ array di dimensione 2^k ed effettuiamo $n/2^{k+1}$ merge, ciascuno di costo al più $c \cdot 2^{k+1}$. Quindi, costo complessivo della generica iterazione k-esima del ciclo più esterno è al più $(c + c \cdot 2^{k+1}) \frac{n}{2^{k+1}} = c\left(1 + 2^{k+1}\right) \frac{n}{2^{k+1}} \leq 2cn$

Ultima domanda: quanti cicli `for` esterni abbiamo? `width` diventa più grande di n dopo $O(\log_2 n)$ iterazioni del ciclo. Ogni ciclo ha costo al più $O(n)$ e abbiamo $O(\log_2 n)$ iterazioni quindi il costo complessivo è $O(n \log_2 n)$. In particolare, dopo i iterazioni del ciclo `for` più esterno `width` vale 2^{i-1} e quindi diventa più grande di n quando $2^{i-1} \geq n$ e ciò accade quando $i - 1 \geq \lceil \log_2 n \rceil$