Introduzione agli algoritmi
Proff. E. Fachini – S. Caminiti
5 maggio 2020

Es. 3: Si consideri il seguente frammento di programma e se ne analizzi il tempo di esecuzione asintotico:

```
fun Test (n) {
        i = n
        while (i > 1) {
                j = i;
                while (j > 0) {
                        k = 0;
                        while (k < n) {
                                k = k + 2;
                        }
                        j = j - 1;
                }
                i = i / 2;
        }
}
```

Il ciclo while più interno è eseguito n/2 volte, quello intermedio i volte, con i uguale a n poi a n/2, …, fino a $n/2^k$, se $2^k \leq n < 2^{k+1}$ e quindi $n/2^k \geq 1$ ma $n/2^{k+1} < 1$ quindi in totale bisogna considerare la somma

$\sum_{i=0,\ldots,k} n/2^i \; n/2 \; \Theta(1) = \Theta(1) \; n^2/2 \sum_{j=0,\ldots,k} 1/2^i \leq \Theta(1) \; n^2/2 \sum_{j=0,\ldots,\infty} 1/2^i$

Sappiamo che $\sum_{j=0,\ldots,\infty} 1/2^i = \Theta(1)$

E quindi $\Theta(1) \; n^2/2 \sum_{j=0,\ldots,\infty} 1/2^i = \Theta(n^2)$

quindi $\sum_{i=0,\ldots,\lg n} n/2^i \; n/2 = O(n^2)$,

poiché il ciclo intermedio viene eseguito la prima volta con i = n e il ciclo più interno viene eseguito n/2 volte, ho che il tempo di esecuzione è anche in $\Omega(n^2)$. In conclusione possiamo affermare che il frammento di codice è eseguito in $\Theta(n^2)$.