Algortimos y Estructuras de Datos I — 2do Parcial

Práctica 4 — Precondición más débil

Skip ———— $wp(skip, Q) \equiv True$

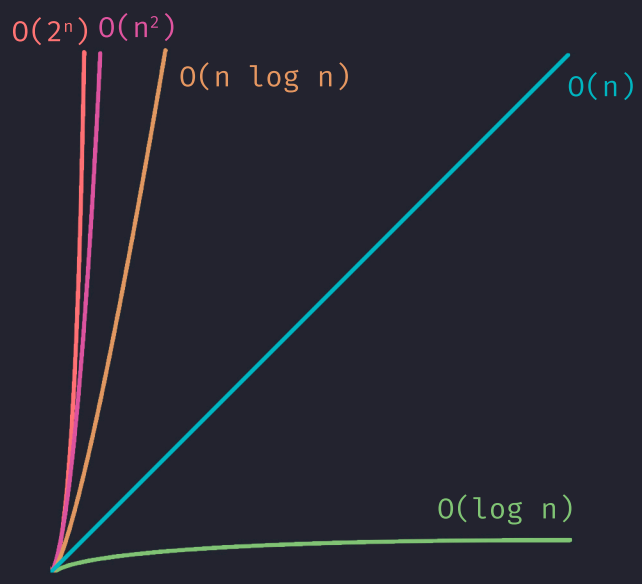
Asignación —— $wp(x := E, Q) \equiv def(x) \wedge_L Q^x_E$

Concatenación — $wp(S1;S2, Q) \equiv wp(S1, wp(S2, Q))$

Condicional —— $wp(\text{if } B \text{ then } S1 \text{ else } S2 \text{ endif}, Q) \equiv def(B) \wedge_L (B \wedge wp(S1,Q) \vee \neg B \wedge wp(S2,Q))$

Práctica 5 — Teorema del Invariante

1) $P_C \Rightarrow I$
2) $\{B \wedge I\}\ S\ \{I\}$
3) $\neg B \wedge I \Rightarrow Q_C$
4) $\{B \wedge I \wedge v_0 = f_v\}\ S\ \{f_v < v_0\}$
5) $I \wedge (f_v \leqslant 0) \Rightarrow \neg B$



Secuencias — Asignación & Indexación

$s[i] := E \equiv setAt(s, i, E)$

$$setAt(s, i, E)[j] = \begin{cases} s[j] & \text{si } j \neq i \\ E & \text{si } j = i \end{cases}$$

Práctica 6 — Testing

```
int f(int x) {
    if (x == 2)   ••
        x++;      •
    return x;     ••
}
```

Test 1 •		Test 2 •	
Input	2	Input	3
Expected	3	Expected	3
Output	3	Output	3
L_C	100%	L_C	66%
B_C	50%	B_C	50%

B_C = 100%

Corrección de programas completos

Sea S un programa con un ciclo C; S1 el programa antes y S2 después de C. Demostremos que $P \Rightarrow wp(S, Q)$

1) $P \Rightarrow wp(S1, P_C)$
2) $P_C \Rightarrow wp(C, Q_C)$ ← Acá probablemente usemos el Teorema del Invariante
3) $Q_C \Rightarrow wp(S2, Q)$

Prácticas 8 y 9 — Complejidad temporal, búsqueda y ordenamiento

Quicksort —— $O(n^2)$
Bubblesort — $O(n^2)$
Insertion —— $O(n^2)$
Selection —— $O(n^2)$
Mergesort —— $O(n \log n)$
Counting —— $O(n+k)$

Binary Search —— $O(\log n)$
Jump Search —— $O(\sqrt{n})$
Linear Search — $O(n)$