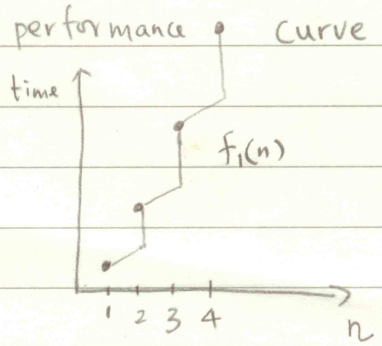
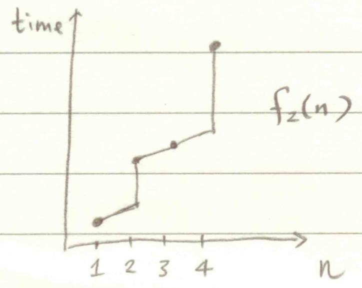
* performance curve



Inc Extend



Dbl Extend

Inc Extend worse than Dbl Extend in time

because $f_1(n) \geq f_2(n)$ for all $n \geq 1$

$f(n) : \mathbb{N} \longrightarrow \mathbb{R}^+ \cup \{0\}$

* writing down $f(n)$

$$f_1(n) = \square + \sum_{i=2}^{n} \left(P \cdot (i-1) + Q \right)$$

$$f_2(n) = \square + \sum_{\substack{2 \le i \le n \\ i = 2^k+1}} \left(P \cdot (i-1) + Q \right) + \sum_{\substack{2 \le i \le n \\ i \ne 2^k+1}} \square$$

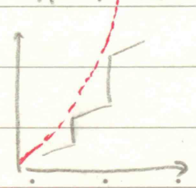
- overly complicated
- $\square$, P, Q platform dependent, not very meaningful
- need: say "approximately"

* approximate upper bound

$$\begin{aligned} f_1(n) &= \square + \sum_{i=2}^{n} \left(P \cdot (i-1) + Q \right) \\ &= \square + \frac{n(n-1)}{2} P + n \cdot Q \\ &\le \square n^2 + \frac{P}{2} n^2 + Q \cdot n^2 \\ &\le \left(\square + \frac{P}{2} + Q \right) n^2 \end{aligned}$$

no worse than

$f_1(n)$ "$\le$" $c \cdot n^2$



* preliminary notation

$f(n) = O(n^2)$ means there exists c such that

$\quad$ ↗ is of

$f(n) \le c \cdot n^2$ for all $n \ge 1$

$\Downarrow$

$f(n) = O(g(n))$ means there exists $c > 0$ such that

big-Oh notation

$f(n) \le c \cdot g(n)$ for all $n \ge 1$

Q: $f_1(n) = O(f_2(n))$?

yes, simply with $c = 1$.

* $f_2(n) = O(\ ?\ )$

1	2	3	4	5	...
$\square$	$P+Q$	$2P+Q$	$\square$	$4P+Q$	

$f_2(n) \le n \cdot \square + n \cdot Q + \underline{\Delta} \cdot P$

$\Delta \to 1 + 2 + 4 + \cdots + 2^{\lfloor \log_2 (n-1) \rfloor}$

$= 2^{\lfloor \log_2 (n-1) \rfloor + 1} - 1$

$\le 2(n-1) - 1$

$\le 2n$

$\le n \cdot (\underbrace{\square + Q + 2P}_{c})$

So

$f_2(n) = O(n)$

* revisit :

getMinPos is of $O(\underbrace{n}_{\text{len}})$ time "linear" time $(P \cdot n + Q)$

Consecutive insert is of $O(1)$ time "constant" time (P)

Seq Search is of $O(n)$ time

Bin Search is of $O(\,?\,)$ time

* Bin Search :

$$f(n) \le \square\left(\lceil \log_2 n \rceil + 1\right) + \triangle$$

$$\overset{?}{\le} \star \cdot \log_2 n$$

?: no when $n = 1$ (because $\log_2 n = 0$)

yes for larger n

$$\square\left(\lceil \log_2 n \rceil + 1\right) + \triangle$$

$$\le \square\left(\log_2 n + 1 + 1\right) + \triangle$$

$$= \square \log_2 n + (2\square + \triangle)$$

$$\le (\square + 2\square + \triangle) \log_2 n \quad \text{for } n \ge 2$$

base is not important for $\log_2 \cdots$ (why?)

* asymptotic notation (slight refinement)

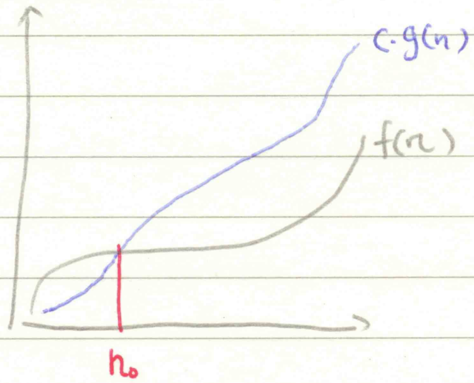
$f(n) = O(g(n))$ meas there exists $c_{>0}$ and n_0 such that

$$f(n) \le c \cdot g(n) \quad \text{for all } n \ge n_0$$

Bin Search is of $O(\log n)$. "logrithmic time"

* meaning of asymptotic notation

- we don't care about "small inputs" usually
  (similarly VERY fast) $n \geq n_0$
- we don't care about "constants"
  (can speedup by platform/hardware) $f(n) \leq c \cdot g(n)$



* OrderedArray.insert

"rotate swap" from back : $O(n)$

Bin Search + cut-in-line : $\boxed{?}$

$O(\log n)$ $\quad$ $O(n)$ $\quad$ $O(n)$

$\downarrow n \geq n_1$ $\quad$ $\downarrow n \geq n_2$

$\leq c_1 \log n$ $\quad$ $\leq c_2 \cdot n$

$\leq c_1 \cdot n$

$\geq \max(n_1, n_2)$

$\leq (c_1 + c_2) \cdot n$

* big-O properties that can be used:

- 封閉律 (closedness)

$$
\begin{array}{lll}
\text{if} & f_1(n) = O(g(n)) & ① \\
+) & f_2(n) = O(g(n)) & ② \\
\hline
\Rightarrow & f_1(n) + f_2(n) = O(g(n)) &
\end{array}
$$

<pf> when $n \geq n_1$, $f_1(n) \leq c_1 \cdot g(n)$ from ①

when $n \geq n_2$, $f_2(n) \leq c_2 \cdot g(n)$ from ②

$\Downarrow$

when $n \geq \underbrace{\max(n_1, n_2)}_{n_0}$, $f_1(n) + f_2(n) \leq \underbrace{(c_1 + c_2)}_{c}\, g(n)$

so $f_1(n) + f_2(n) = O(g(n))$

- 遞移律 (transitivity)

$$
\begin{array}{lll}
\text{if} & f_1(n) = O(g_1(n)) & \\
 & g_1(n) & = O(g_2(n)) \\
\Rightarrow & f_1(n) & = O(g_2(n))
\end{array}
$$

<pf> similar to the above as exercise

- use the two laws above

⓪ BinSearch + cut-in $= O(n)$

① $a_m n^m + a_{m-1} n^{m-1} + \cdots + a_0 n^0 = O(n^m)$

- $\underbrace{\text{for-loop}}_{f_1(n)} \times \underbrace{\text{inner}}_{f_2(n)}$? see HW2

組合

* using big-O

$f(n) = O(\log n)$ implies

$f(n) = O(n)$ and $f(n) = O(\log n + \log\log n)$

$O(n^2)$

$O(n^m)$

$O(2^n)$

prefer tightest and simplest

* other asymptotic notations

O: upper bound

Ω: lower bound

$f(n) = \Omega(g(n))$ iff $\exists\, c > 0,\ n_0 > 0$ s.t.

$f(n) \geq c \cdot g(n)$ for all $n \geq n_0$

Θ : O and Ω

* "asymptotic"

$\boxed{\text{if } \lim_{n\to\infty} \frac{f(n)}{g(n)} = \square \text{ exists}}$. $f(n) = O(g(n))$

$\Downarrow$ $\Uparrow$

$\boxed{\text{for all } \epsilon > 0 \text{ exists } n_\epsilon \text{, s.t. } \left| \frac{f(n)}{g(n)} - \square \right| < \epsilon \text{ for all } n \geq n_\epsilon}$ $\Rightarrow$ let $\epsilon = 0.1126$

$f(n) < (\overbrace{\square + 0.1126}^{c})\, g(n)$

for all $n \geq \underbrace{n_{0.1126}}_{n_0}$