2-6 Algorithmic Methods

Hengfeng Wei

hfwei@nju.edu.cn

April 07, 2020

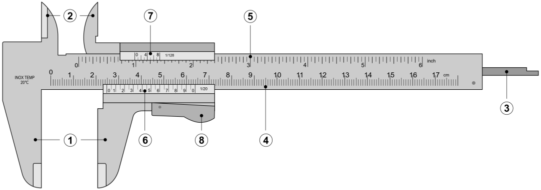
$$T(n) = aT(n/b) + f(n) \quad (a > 0, b > 1)$$

Assume that $T(n)$ is constant for sufficiently small n.

$f(n)$ is asymptotically positive.

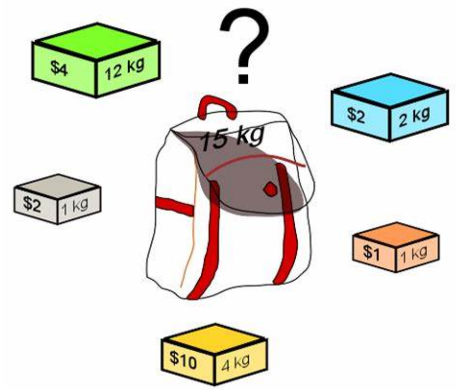
$$\boxed{T(1) = 0 \textit{ vs. } T(1) = d \neq 0}$$

Convex Polygon Diameter



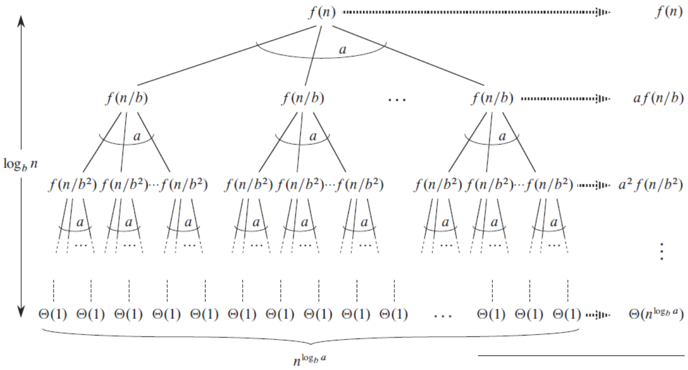
Correctness Proof

Integer Knapsack



Algorithm & Time Complexity

$$T(n) = aT(n/b) + f(n)$$



$$T(n) = n^{\log_b a} T(1) + \sum_{j=0}^{\log_b n - 1} a^j f\left(\frac{n}{b^j}\right)$$

$$T(n) = \Theta(n^{\log_b a}) + \sum_{j=0}^{\log_b n - 1} a^j f\left(\frac{n}{b^j}\right)$$

$$\boxed{f(n) \ v.s. \ n^{\log_b a}}$$

$$f(n) = O(n^{\log_b a - \epsilon}) \implies T(n) = \Theta(n^{\log_b a})$$

$$f(n) = \Theta(n^{\log_b a}) \implies T(n) = \Theta(n^{\log_b a} \log n)$$

$$f(n) = \Omega(n^{\log_b a + \epsilon}) \implies T(n) = \Theta(f(n))$$

$$T(n) = n^{\log_b a} T(1) + \sum_{j=0}^{\log_b n - 1} a^j f\left(\frac{n}{b^j}\right)$$

$$a^{\log_b n - 1} f\left(\frac{n}{b^{\log_b n - 1}}\right) = a^{\log_b n - 1} f(b) = a^{\log_b n} \frac{f(b)}{a}$$

$$\begin{aligned} T(n) &= n^{\log_b a} T(1) + \sum_{j=0}^{\log_b n - 1} a^j f\left(\frac{n}{b^j}\right) \\ &= n^{\log_b a} \left(\frac{f(b^{\log_b n})}{a^{\log_b n}} + \ldots + \frac{f(b^2)}{a^2} + \frac{f(b)}{a} + T(1) \right) \end{aligned}$$

$f(n)$ is asymptotically positive.

$$T(1) = d = \Theta(1), \quad (d \text{ can be } 0)$$

$$T(n) = n^{\log_b a} T(1) + \sum_{j=0}^{\log_b n - 1} a^j f\left(\frac{n}{b^j}\right)$$

$$f(n) = O(n^{\log_b a - \epsilon}) \implies T(n) = \Theta(n^{\log_b a})$$

$$f(n) = \Theta(n^{\log_b a}) \implies T(n) = \Theta(n^{\log_b a} \log n)$$

$$f(n) = \Omega(n^{\log_b a + \epsilon}) \implies T(n) = \Theta(f(n))$$

$$T(n) = n^{\log_b a} \left(\frac{f(b^{\log_b n})}{a^{\log_b n}} + \ldots + \frac{f(b^2)}{a^2} + \frac{f(b)}{a} + T(1) \right)$$

$f(n)$ is asymptotically positive.

$$\sum_{j=0}^{\log_b n - 1} a^j f\left(\frac{n}{b^j}\right) = \Omega(n^{\log_b a})$$

$$T(n) = aT(n/b) + f(n)$$

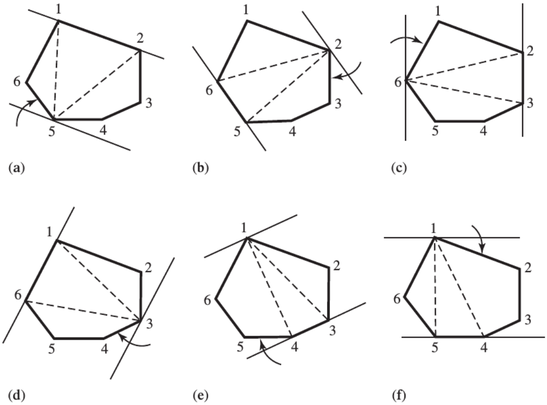
$$T(n) = n^{\log_b a} \left(\frac{f(b^{\log_b n})}{a^{\log_b n}} + \ldots + \frac{f(b^2)}{a^2} + \frac{f(b)}{a} + T(1) \right)$$

What if $f(n) = 0$?

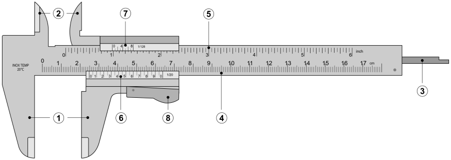
$$T(n) = n^{\log_b a} T(1)$$

$$T(n) = \begin{cases} 0, & T(1) = 0 \\ \Theta(n^{\log_b a}), & T(1) = d \neq 0 \end{cases}$$

Convex Polygon Diameter



Rotating Caliper



"Computational Geometry"
Ph.D Thesis, Michael Shamos, 1978

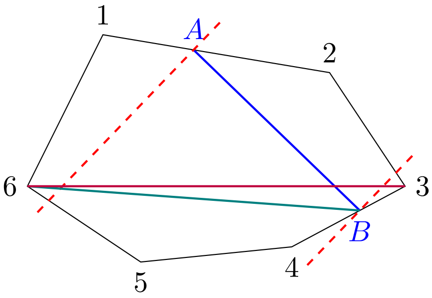
"Solving Geometric Problems with the Rotating Calipers", 1983

Correctness

WAIT BUT WHY

Theorem (DH 4-8)

If AB is a diameter of a convex polygon P, then A and B are vertices.



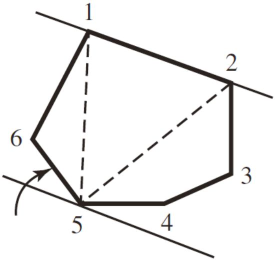
BUT, we have *not* enumerated *all* pairs of vertices.

We have enumerated *all antipodals.*

Definition (Line of Support)

A line L is a *line of support* of a convex polygon P if

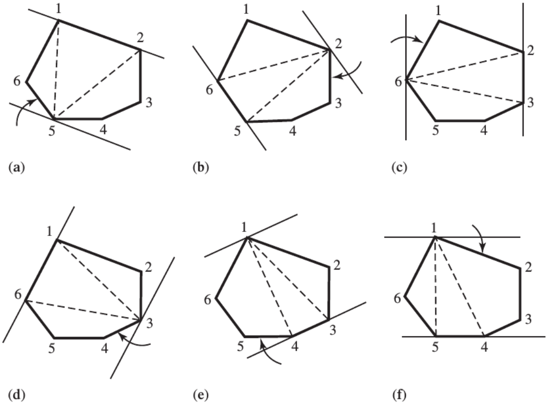
$$L \cap P = \text{ a vertex/an edge of } P.$$



$L \cap P \neq \emptyset$ $\quad\quad$ P lies entirely on one side of L.

Definition (Antipodal)

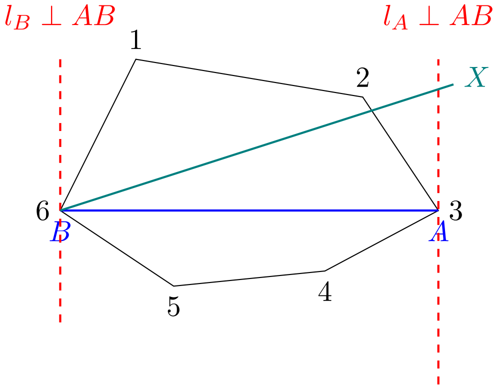
An *antipodal* is a pair of points that admits parallel supporting lines.



We have enumerated *all* antipodals by *rotating* through all angles.

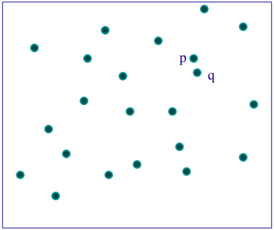
Theorem (We Won't Miss the Diameter)

If AB is a diameter of a convex polygon P, then AB is an antipodal.



$L \cap P \neq \emptyset$ P lies entirely on one side of L.

Finding the Closest Pair of Points (Additional: DH 4-10)



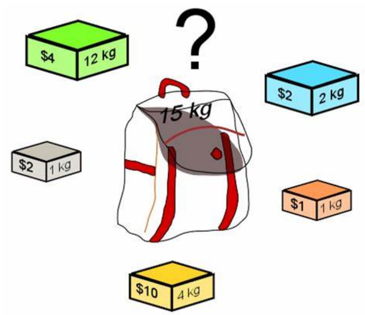
A Classical and Beautiful Divide-Conquer-Combine Algorithm:

Section 33.4, CLRS

DH 4.13 (Integer Knapsack)

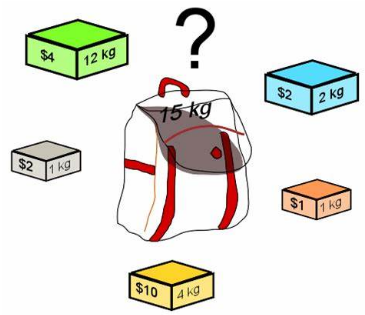
$$N = 5$$
$$Q = [3, 1, 4, 5, 1] \quad (quantity)$$
$$W = [10, 20, 20, 8, 7] \quad (weight)$$
$$P = [17, 42, 35, 16, 15] \quad (profit)$$

$$C = 103$$



0-1 Knapsack

$$\forall i : Q[i] = 1$$



DH 4.13 (Integer Knapsack)

$$
\begin{aligned}
N &= 5 \\
Q &= [3, 1, 4, 5, 1] \quad (\textit{quantity}) \\
W &= [10, 20, 20, 8, 7] \quad (\textit{weight}) \\
P &= [17, 42, 35, 16, 15] \quad (\textit{profit})
\end{aligned}
$$

$$
\begin{aligned}
N' &= \sum_i Q[i] \\
W' &= [\ldots, \underbrace{W_i}_{\#=Q_i}, \ldots] \\
P' &= [\ldots, \underbrace{P_i}_{\#=Q_i}, \ldots]
\end{aligned}
$$

$K[c, i]$:

The maximal profit obtained using knapsack of capacity c with items of $x_1 \dots x_i$.

$K[C, N]$

$\boxed{\text{Using the item } x_i \text{ or not?}}$

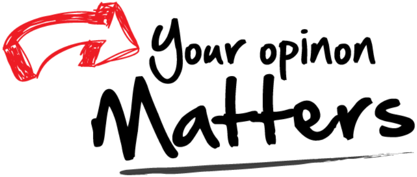
$$K[c,i] = \max \begin{cases} K[c, i-1], \\ K[c - W[i], i-1] + P[i], \quad W[i] \le c \end{cases}$$

$$\text{Time complexity} : \Theta(NC)$$

Is this a polynomial algorithm?


KEEP
CALM
AND
STAY
TUNED


Thank You!



Office 302

Mailbox: H016

hfwei@nju.edu.cn