Project discussion

March 15, 2017

Sorting Problem

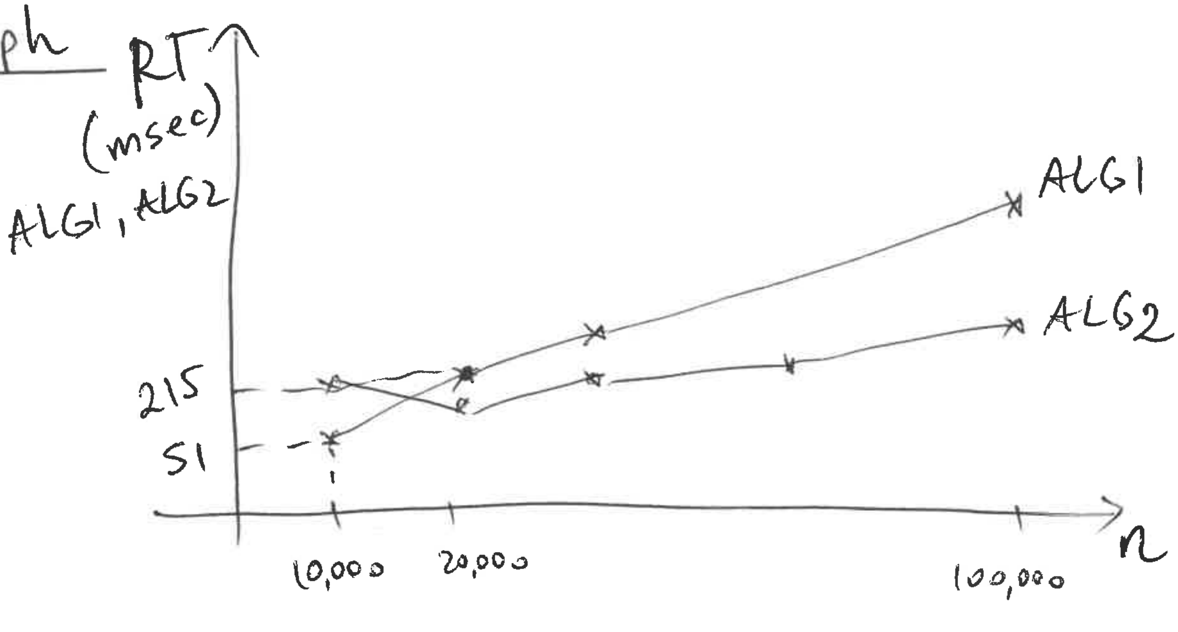
ALG1: Bubble-Sort, worst-case RT = $O(n^2)$
ALG2: Merge-Sort, worst-case RT = $O(n \cdot \lg n)$
↳ Divide-and-Conquer

input size: n – no. of elements

$n = 10{,}000 ; 20{,}000 ; \ldots ; 100{,}000$

number of runs $m = 10$
Data structure: array
Generate the input values: using random()

Graph



* Run the 2 algs on exactly the same input

Theoretical RT vs. Empirical RT

ALG1 : worst-case $RT = O(n^2)$ $\Rightarrow$ $RT \le c n^2$

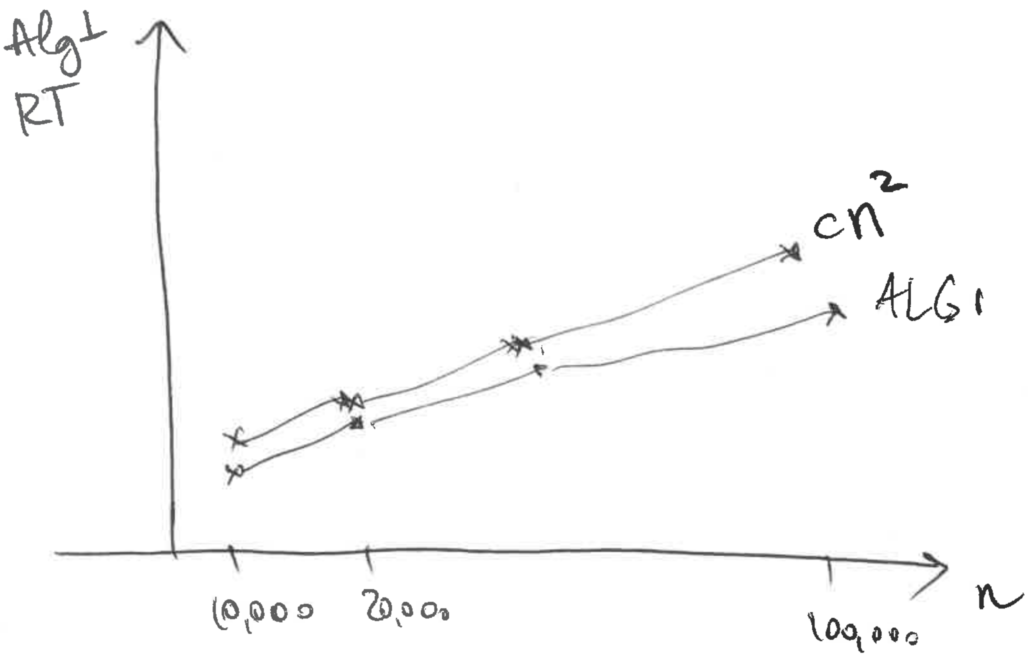
n	Empirical RT (msec)	Theoretical RT	const c approximation
10,000	51	10^8	$c_1 = 51 \cdot 10^{-8}$
20,000	215	$4 \cdot 10^8$	$c_2 = \frac{215}{4} \cdot 10^{-8}$
⋮	⋮	⋮	⋮
100,000		$100 \cdot 10^8$	c_{10}

$\frac{\text{Empirical RT}}{\text{Theoretical RT}}$

$c = \max(c_1, c_2, \ldots, c_{10})$

$\boxed{c =\quad}$

Note: if needed, ignore the first run of the alg.



* repeat for ALG2

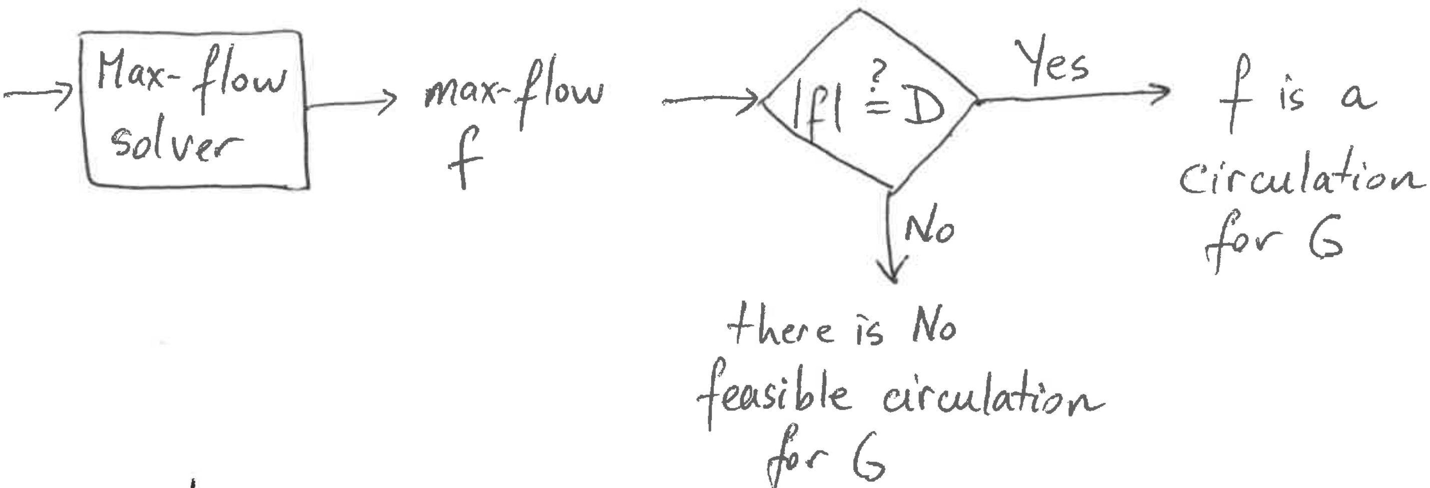
Circulations w/ demands

Input

- graph $G(V,E)$ directed
- each edge has a capacity c
- each vertex v has a demand d_v

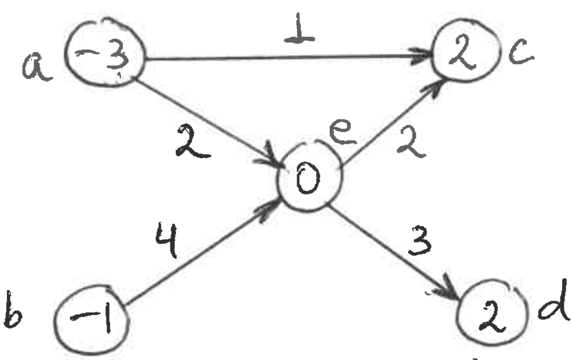
Flow Network

- graph $G'(V',E')$ directed
- add super-source s^* and super-sink t^*
- keep edges in E and their capacities
- add edges from s^* to sources (s^*,v) with capacity $-d_v$
- add edges from sinks to t^* (v,t^*) with capacity d_v



example

Consider the graph $G(V,E)$ with the following capacity and demand constraints:

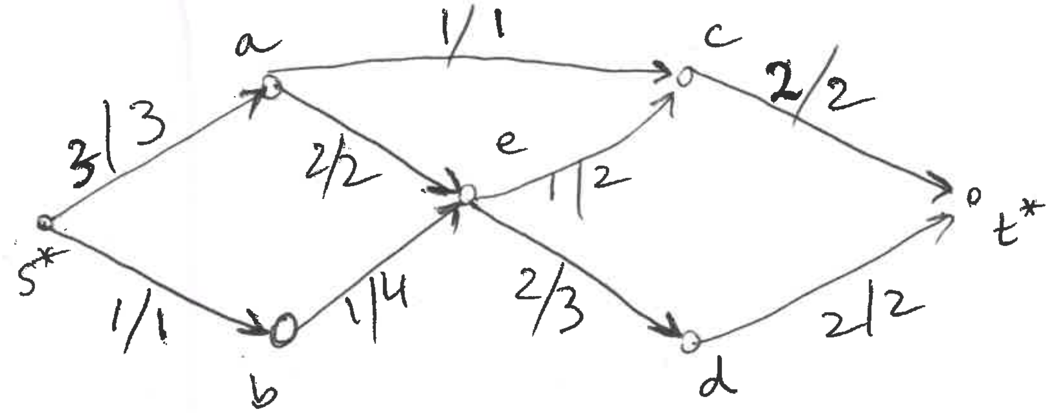


Find a feasible circulation.

Solution

graph G' (flow network G')

$D = 4$



$p = \langle s^*, a, e, d, t^* \rangle$ $c_f(p) = 2$

$p = \langle s^*, b, e, c, t^* \rangle$ $c_f(p) = 1$

$p = \langle s^*, a, c, t^* \rangle$ $c_f(p) = 1$

$|f| = 4 = D \Rightarrow G$ has a feasible circulation: