November 10, 2023   10:43 AM

Approx Distance Oracles

Problem Given undir. nonnegative weighted graph $G=(V,E)$,
build static data structure s.t.
given $u,v \in V$, ~~find $d(u,v)$~~

Shortest-path distance.

find value $\tilde{d}$ s.t.

$$d(u,v) \le \tilde{d} \le c\, d(u,v)$$

approx factor

Exact: APSP $\Rightarrow$
- preproc time $O(n(n\log n + m))$
- space $O(n^2)$
- query time $O(1)$

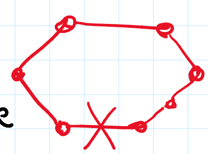
Approx: $(1+\varepsilon)$-approx APSP preproc $O(n^{2.38})$ (Zwick '99)
3-approx APSP $\tilde{O}(n^2)$ (Cohen-Zwick '01)

better data structures? Subquadratic space?

e.g. Spanners
$\exists$ subgraph of size $O(n^{1+1/k})$
that approximate all dists with factor $2k-1$

tight assuming Erdős girth conjecture ($\exists$ unwtd undir graph with $\Omega(n^{1+1/k})$ edges & girth $2k+1$)

$\Rightarrow$ good space bd
but not necessarily good query time



Thorup-Zwick '01:
- preproc time $\boxed{O(k m n^{1/k})}$
- space $\boxed{O(k n^{1+1/k})}$
- query time $\boxed{O(k)}$
- approx factor $\boxed{2k-1}$

$n^{\frac{1}{\log n}} = O(1)$

e.g. 3-approx, space $O(n^{3/2})$, query $O(1)$
5-approx, $n^{4/3}$

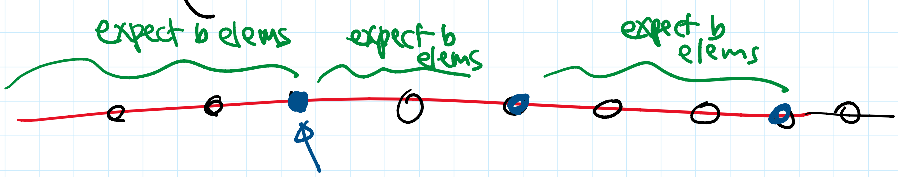
e.g. 3-approx, space $O(n^{3/2})$, query $O(1)$
5-approx, $n^{4/3}$
⋮
199-approx, $n^{1.01}$
$O(\log n)$-approx, space $O(n \log n)$, query $O(\log n)$

Thorup-Zwick Method

idea: random sampling

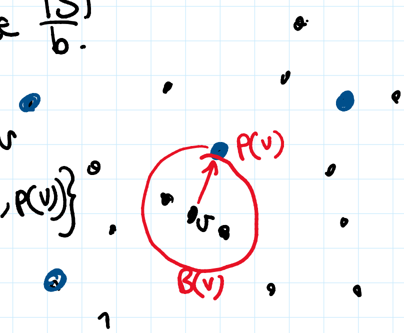
Fact Given a set A of n numbers,
take a rand. subset $A' \subseteq A$ of $\frac{n}{b}$ numbers.

$$E\left(\left|\{x \in A - A' : x < \min(A')\}\right|\right) = O(b).$$



preproc(S): // given $S \subseteq V$ (initially, $S = V$)

S' = rand subset of S of size $\frac{|S|}{b}$.
for each $v \in V$
$p(v)$ = nearest vertex in S' to v
$B(v) = \{x \in S - S' : d(v,x) < d(v,p(v))\}$



expected size of $B(v)$ is $O(b)$

store $d(v, p(v))$
and $d(v,x)$ $\forall x \in B(v)$.

preproc(S').

Set $b = n^{1/k}$. $\Rightarrow$ hierarchy with k levels
$(|S| = n \to n^{1-1/k} \to n^{1-2/k} \to \cdots)$
$\Rightarrow$ expected space $O(bn \cdot k) = \boxed{O(kn^{1+1/k})}$

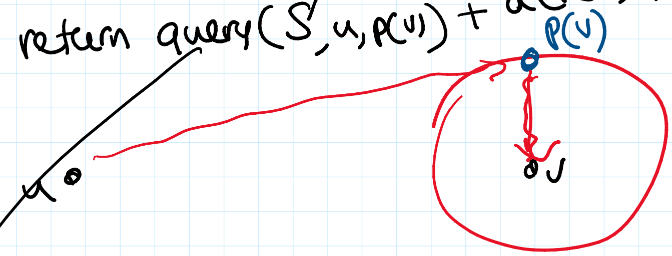
How to answer queries?

How to answer queries?

First Attempt: query(S, u, v): // given $u \in V$, $v \in S$

if $u \in B(v)$ return $d(u,v)$

else return query$(S', u, p(v)) + d(p(v), v)$



if query$(S', -)$ has approx factor c,

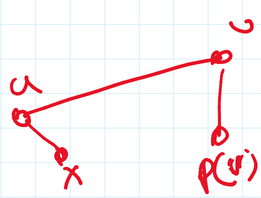
$$
\begin{aligned}
&\le c\, d(u, p(v)) + d(p(v), v) \\
&\le c\big(d(u,v) + d(v, p(v))\big) + d(p(v), v) \\
&= c\, d(u,v) + (c+1)\, \underbrace{d(v, p(v))}_{\le d(v,u)} \\
&= (2c+1)\, d(u,v)
\end{aligned}
$$

$\xRightarrow{\text{exp blow-up}}$ final approx factor $2^k - 1$

Better alg'm: idea - alternately look around $B(u)$ and $B(v)$

query(S, u, v, x): (initially, $S = V$, $x = u$, $\alpha = 0$.)

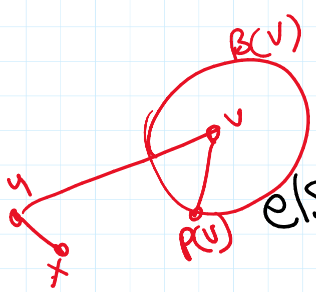
// given $u, v \in V$ and a "pivot" vertex $x \in S$ which is "close" to u, i.e. $\boxed{d(u,x) \le \alpha\, d(u,v)}$
& assume $d(u,x)$ is stored



if $\underline{x \in B(v)}$ then // $d(u,x)$ is stored & $d(x,v)$ is stored

return $d(u,x) + d(x,v)$

$$
\begin{aligned}
&\le d(u,x) + d(x,u) + d(u,v) \\
&= (2\alpha + 1)\, d(u,v) \Rightarrow \text{approx factor } 2\alpha+1.
\end{aligned}
$$



else return query$(S', v, u, p(v))$

// $d(v, p(v)) \le d(v, x)$

$p(v)$ else return query(…)

x

// $d(v, p(v)) \le d(v, x)$
$\le \cancel{d(u,x)} + d(u,v)$
$\le \boxed{(\alpha+1)\, d(u,v)}$

$\Rightarrow$ max $\alpha = k-1$

$\Rightarrow$ approx factor $2(k-1)+1 = \boxed{2k-1}$.

Rmks - can be derand.
So far, method works for any metric space!
not just graphs.

Preproc time?

- computing $p(v)$ $\forall v \in V$ by Dijkstra
- Computing $B(v)$ $\forall v \in V$. by modified Dijkstra ... (skipping details)

$\Rightarrow O(bm \cdot k) = \boxed{O(kmn^{1/k})}$

Rmks - cond. lower bds (Jin-Xu '22 / Abboud et al '22)
2k-approx $\Omega(m^{1+1/2k})$ ...

- below-3 approx
  Patrascu-Roditty '10: 2-approx, $O(m^{1/3} n^{4/3})$ space
  ⋮
- DOs for diff. types of graphs ...

STRINGS

Motivating Problem Given "text" string $T = t_1 t_2 \cdots t_n \in \Sigma^*$,
build a static data structure s.t.
given "pattern" string $P = p_1 p_2 \cdots p_m$, $(m \le n)$
decide whether P is a substring of T

(if so, report one occurrence
or all occurrences
or count)

e.g. $T = 01011011010$

$P = 1011$

Non-DS version: (standard string matching problem)

trivial alg'm $O(mn)$ time

Knuth-Morris-Pratt '77 $O(n)$ time

(many other $O(n)$ alg'ms...
e.g. Karp-Rabin '87 in CS473)

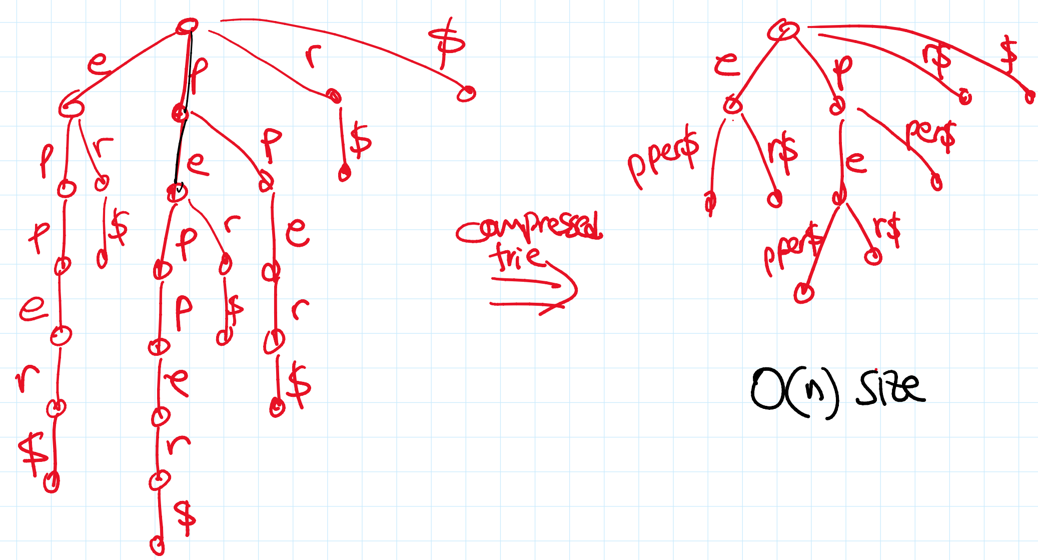
DS version?

Suffix Tree

compressed trie for all suffixes of T.

e.g. T = "pepper$"

1 pepper$
2 epper$
3 pper$
4 per$
5 er$
r$
$



O(n) size

P = "epp"
"pe"

query $O(m)$

or $O(m + \#occ)$.