6.S899 Distributed Graph Algorithms Lecture 11 MIT 12/05/2014

Material covered:

Deterministic all pairs shortest paths on weighted graphs in linear time. This can be found in section 4.1 of "Stephan Holzer and Roger Wattenhofer. Optimal distributed all pairs shortest paths and applications, Proceedings of the 31st annual ACM Symposium on Principles of Distributed Computing (PODC'12), pages 355-364". We also covered an animation of this algorithm by Jukka Suomela: http://users.ics.aalto.fi/suomela/apsp/

Optimal deterministic source detection. This can be found in section 4 of "Christoph Lenzen and David Peleg, efficient distributed source detection with limited bandwidth, Proceedings of the 32nd annual ACM Symposium on Principles of Distributed Computing (PODC'13), pages 375-382".
We follow the presentation of Christoph Lenzen, lecture notes 8 "Distance Approximation and Routing" of a class on Theory of Distributed Systems at MPI Saarbruecken: http://resources.mpi-inf.mpg.de/departments/d1/teaching/ws14/ToDS/script/routing.pdf

Fast deterministic computation of small k-dominating sets. This follows the material in section 2 of "Shay Kutten and David Peleg, Fast distributed construction of k-dominating sets and applications, Proceedings of the fourteenth annual ACM symposium on Principles of Distributed Computing (PODC'95), pages 238-251, 1995".

Deterministic (1+eps)-approximation of the diameter of a unweight graph. This can be found in section 6.2 of "Stephan Holzer and Roger Wattenhofer. Optimal distributed all pairs shortest paths and applications, Proceedings of the 31st annual ACM Symposium on Principles of Distributed Computing (PODC'12), pages 355-364". However, in this lecture we use the source detection algorithm mentioned above instead of the S-SP algorithm.

LAST LECTURE:

- MIN-CUT

THIS LECTURE:

DETERMINISTIC, UNWEIGHTED GRAPHS

- ALL-PAIRS SHORTEST PATHS $O(n)$.
- (S,H,K)-SOURCE DETECTION IN $H+K+1$.
- k-DOMINATING SET OF SIZE $\lfloor n/k \rfloor$ IN $O(D)$
- $(1+\varepsilon)$-APPROXIMATION OF D IN TIME $O(n/D + D)$

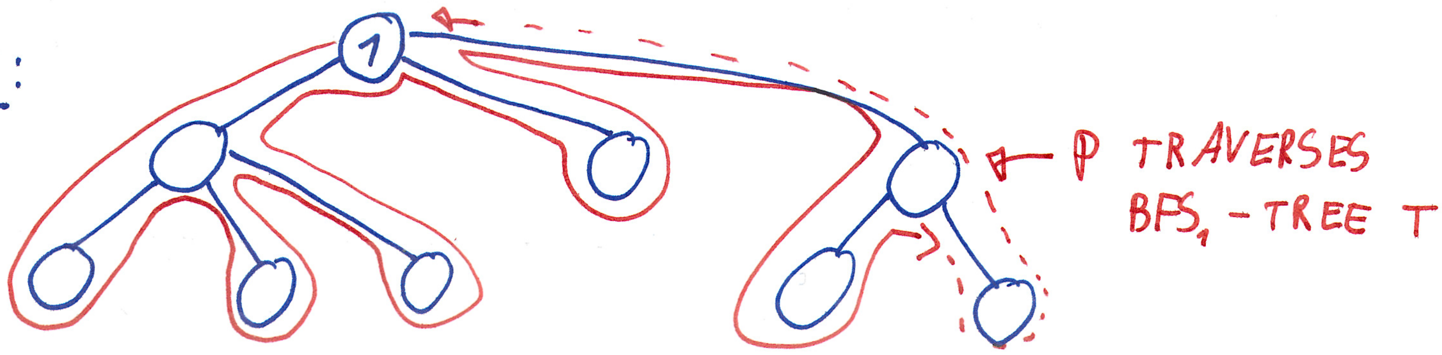
"PIPELINING" TECHNIQUES.

DEF: BFS_v DENOTES BFS-TREE / COMPUTATION STARTED IN $v \in V$.

ALGO: (APSP) // ASSUME LEADER 1 IS GIVEN

COMPUTE BFS_1-TREE T ROOTED IN 1;
SEND A TOKEN/PEBBLE P TO TRAVERSE T IN DFS-WAY;
WHILE P TRAVERSES T DO
    IF P VISITS A NODE v NOT VISITED BEFORE
    THEN
        WAIT ONE TIME SLOT;
        START BFS_v-COMPUTATION FROM NODE v;
            // COMPUTES ALL DISTANCES TO v.

EX:



<u>LEM I</u>: IN THE ALGORITHM NO NODE w IS SIMULTANEOUSLY ACTIVE FOR BOTH BFS_u AND BFS_v.

<u>PROOF</u>: CONSIDER BFS_u / BFS_v STARTED AT TIME $t_u > t_v$ AT NODES $u \neq v \in V$.

(1) $\Rightarrow$ P VISITED v AFTER u $\Rightarrow t_v \geq t_u + d(u,v) + 1$ (1)

AS P TRAVERSES T $\to d(u,v)$

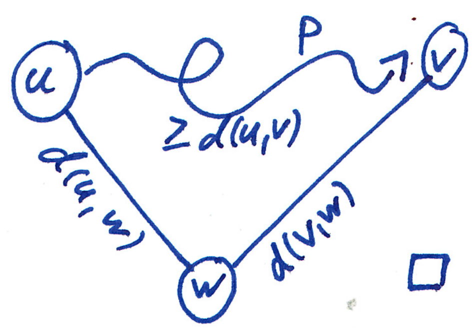
P WAITS 1 SLOT $\to +1$

(2) CONSIDER $w \neq u,v$ INVOLVED IN BFS_u / BFS_v AT TIME $t_u + d(u,w)$ / $t_v + d(v,w)$ IF EXECUTED INDEPENDENTLY.

IT IS: $t_v + d(v,w) \overset{(1)}{\geq} (t_u + d(v,u) + 1) + d(v,w)$

$\geq t_u + d(u,w) + 1$

$\neq t_u + d(u,w)$



$\square$

<u>THM</u>: THE ALGORITHM COMPUTES APSP IN $O(n)$.

<u>PROOF</u>: LEM I IS VALID FOR ANY $\underbrace{u \neq v}_{+} \neq w \in V$.

$\Rightarrow$ ALL MESSAGES CAN BE SENT WITHOUT CONGESTION.

$\Rightarrow$ EACH BFS_v STOPS AFTER D STEPS.

$\Rightarrow$ ~~RUNTIME~~ P TRAVELS FOR $O(n)$ STEPS

$\Big\}\ O(n + D) = O(n)$

(2) $\square$

REM: • APSP $\in \Omega(n)$, SO THIS ALGO IS OPTIMAL.

• WHAT IF WE WANT TO COMPUTE DISTANCES BETWEEN S AND V, $S \subseteq V$? CAN WE DO $O(|S|+D)$?

DEF: TOTAL ORDER OF DISTANCE/NODE-PAIRS

MORE GENERAL $\rightsquigarrow$ GOOD FOR APPLICATIONS

$(\underset{\in \mathbb{N}_0}{d_v}, \underset{\in V}{v}) < (\underset{\in \mathbb{N}_0}{d_w}, \underset{\in V}{w})$ IFF
- "$d_v < d_w$", OR
- "$d_v = d_w$ AND $v < w$".

DEF: GIVEN SOURCES $S \subseteq V$, HOP-BOUND H, SOURCE-BOUND K,

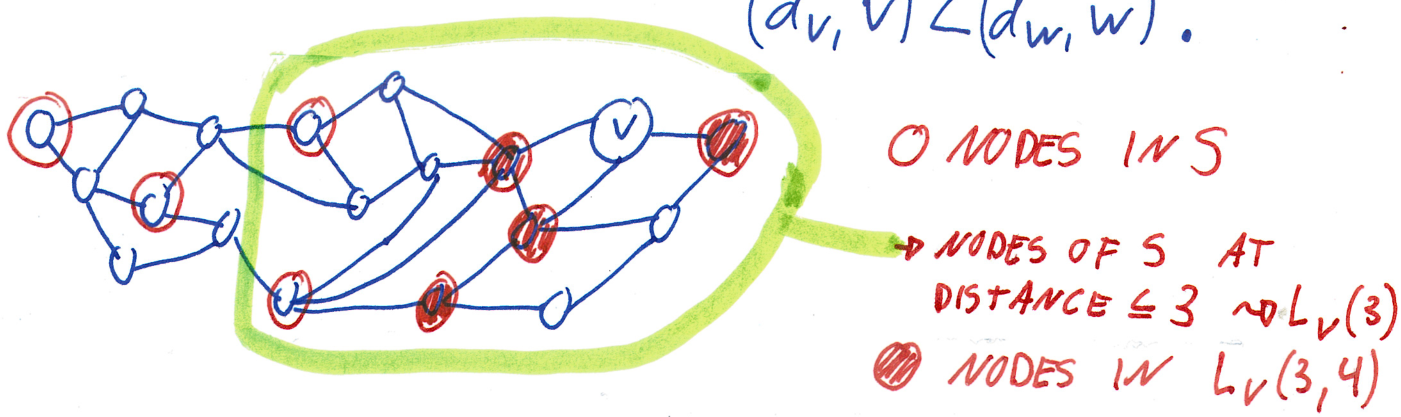
$L_v := \{(d(v,s),s) \mid s \in S\}$

$L_v(H) := \{(d(v,s),s) \in L_v \mid d(v,s) \leq H\}$

$L_v(H,K) :\subseteq L_v(H)$ s.t.
- $|L_v(H,K)| = \min\{K, |L_v(H)|\}$
- FOR EACH $(d_w,w) \in L_v(H) \setminus L_v(H,K)$ AND $(d_v,v) \in L_v(H,K)$ IT IS $(d_v,v) < (d_w,w)$.

"THE SMALLEST K ENTRIES OF $L_v(H)$".

EX:



DEF: (S,H,K)-*detection* PROBLEM IS TO COMPUTE $L_v(H,K)$ FOR EACH $v \in V$.

APPLICATIONS:
- DISTANCE ORACLES
- COMPACT ROUTING
- STEINER-TREE APPROX.
- (WEIGHTED) APSP APPROX.

ALGO: USE A BFS TREE TO ASSIGN NEW IDs IN $\{1,\dots,|S|\}$ TO NODES IN S; // AGGREGATE + DIVIDE VALUES

FOR $i=1,\dots,|S|$ DO

NODE WITH NEW ID i STARTS BFS-COMPUTATION. STOP AFTER H ROUNDS.

THM: COMPUTES $(S,H,|S|)$-DETECTION IN $O(D+|S|\cdot H)$ TIME.

REM: COMPUTING APSP WITH ALGO I IN $O(n)$ MIGHT BE FASTER.

ALGO: PIPELINED BELLMAN-FORD:

$$L_v := \begin{cases} \emptyset & : v \notin S \\ \{(0,v)\} & : v \in S \end{cases};$$ // WILL BE UPDATED, BUT KEEPS ONLY SMALLEST COPY OF (d_s, s) PER NODE s

FOR $r=1,\dots,H+K+1$ DO

$L'_v := \{a \in L_v \mid a \text{ NOT SENT BEFORE}\}$;

IF $L'_v \neq \emptyset$ THEN

$(d_s, s) = \min(L'_v)$;

SEND (d_s+1, s) TO NEIGHBORS;

FOR EACH (d_s, s) RECEIVED FROM NEIGHBORS DO

$L_v := L_v \cup \{(d_s, s)\}$;

RETURN L_v;

THM: THE ALGORITHM SOLVES THE (S,H,K)-DETECTION PROBLEM IN TIME $H+K+1$.

PROOF: **RUNTIME:** $H+K+1$ ITERATIONS, EACH TIME 1.

CORRECTNESS: FEW MORE LEMMATA.

DEF: L_v^r := CONTENT OF L_v IN ROUND r

LEM: $(d_w, w) \in L_v^r$ IMPLIES 1) $w \in S$, 2) $d_w \geq d(v,w)$.

PROOF: 1) ONLY THESE ENTRIES WITH $w \in S$ ARE EXCHANGED.

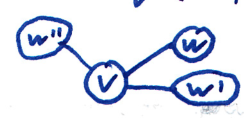
2) d_w-VALUE WAS INCREASED BY 1 IN EACH HOP.

$\square$

REM I: 1) $r' < r \Rightarrow L_v^{r'} \subseteq L_v^r$ W.R.T. UPDATES $(\bullet, S)$

2) $L_v(h,k) \subseteq L_v^r \Rightarrow$ K SMALLEST ELEMENTS OF L_v^r ARE $L_v(H,K)$.

LEM II: LET $h \leq H$, $k \leq K$, $v \in V$, THEN:

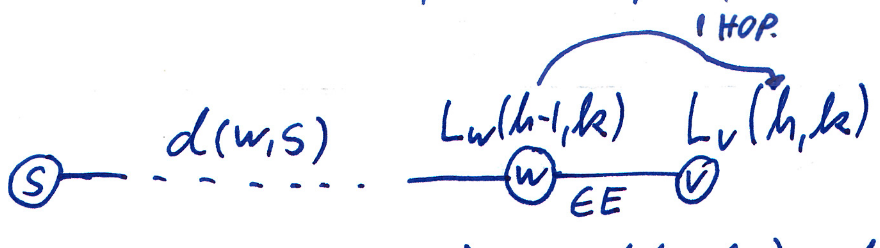


$$L_v(h,k) \subseteq \left\{ (d(w,s)+1, s) \;\middle|\; \begin{array}{l} (d(w,s),s) \in L_w(h-1,k) \\ \text{AND } (v,w) \in E \end{array} \right\}.$$

DENOTE THE PART CONTRIBUTED BY AS FIXED $w \in \Gamma(v)$ BY

$=: L_{w,v}(h-1,k)$

PROOF:



FOR ANY $(d(v,s),s) \in L_v(h,k)$ AND $w \in \Gamma(v)$ ON A SHORTEST s-v-PATH, IT IS

$$\left.\begin{array}{rl} d(w,s) &= d(v,s) - 1 \\ &\leq h \quad\; - 1 \end{array}\right\} \text{AS } (d(v,s),s) \in L_v(h,k)$$

$\Rightarrow (d(w,s),s) \in L_w(h-1)$.

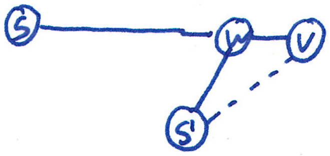
ASSUME $(d(w,s),s) \notin L_w(h-1,k)$

$\Rightarrow$ THERE ARE k ELEMENTS $(d(w,s'),s') \in L_w(h-1)$

S.T. $(d(w,s'),s') < (d(w,s^*),s^*)$.

$\Rightarrow ((d(v,s'),s') < (d(v,s),s)$

AS w IS ON SHORTEST (v,s)-PATH.



FURTHERMORE $(d(v,s'),s') \in L(h)$, AS

$d(v,s') \le d(v,s) \le h$

$\Rightarrow$ THERE ARE k ELEMENTS IN $L(h)$ SMALLER THAN $(d(v,s),s)$

$\lightning$ DEF. OF $L_v(h,k)$. $\square$

<u>LEM</u>: LET $v \in V$, $r \in \{0,\dots,H+K+1\}$, $h+k \le r+1$, THEN IT IS

(1) $L_v(h,k) \subseteq L_v^r$, AND

(2) v HAS SENT $L_{w',v}(h,k)$ BY END OF ROUND $r+1$ TO EACH $w' \in \Gamma(v)$.

<u>PROOF</u>: INDUCTION ON r.

<u>$r=0$, I.E. $h+k=1$:</u> CASE $h=0$: IT IS $L_v(0,k) = \begin{cases} \emptyset & : v \notin S \\ \{(0,v)\} & : v \in S \end{cases}$

FOR ALL $k \in \{0,1\}$ AND $= L_v^0$.

CASE $h=1$: $\Rightarrow k=0 \Rightarrow L_v(h,0) = \emptyset \subseteq L_v^0$

THIS PROVES (1). AS $|L_v(h,k)| \le 1$, $L_v(h,k)$ WILL BE SENT IN ROUND $r+1$, WHICH PROVES (2).

<u>$r \to r+1$, I.E.:</u> ASSUME (1) AND (2) FOR $h'+k' = r+1$.

WE PROVE (1) AND (2) FOR $h+k = r+2$.

CASE $h=0$: $L_v(0,k) = \begin{cases} \emptyset & : v \notin S \\ \{(0,v)\} & : v \in S \end{cases}$ FOR $k \in \{0, \dots, r+2\}$

$\subseteq L_v^{r+1}$.

CASE $h > 0$: <u>WE SHOW STATEMENT (1):</u>

FOR $h'+k' \leq r+1$, v'S NEIGHBORS $\{w\}$ EACH SENT $L_{v,w}(h',k')$ TO v IN ROUND $r+1$.

$\Rightarrow$ v KNOWS $L_v(h'+1,k')$ BY LEM II.

SIMILARLY v RECEIVED $L_{v,w}(h'-1,k'+1)$

$\Rightarrow$ v KNOWS $L_v(h',k'+1)$ BY LEM II.

THEREFORE IT IS

$$L_v(h,k) \subseteq L_v(h'+1,k') \cup L_v(h',k'+1)$$

AS $h'+k'+1 = h+k$

$$\subseteq \bigcup_{w \in \Gamma(v)} \left(L_{v,w}(h',k') \cup L_{v,w}(h'-1,k')\right)$$

$$\subseteq L_v^{r+1}$$

$\Rightarrow$ STATEMENT (1)

<u>WE NOW SHOW STATEMENT (2):</u>

- AFTER ROUND $r+1$, v KNOWS $L_v^*(h,k)$. THIS FOLLOWS FROM WHAT WE JUST PROVED.
- BY INDUCTION HYPOTHESIS, STATEMENT (2), $L_v(h',k')$ IS ALREADY SENT BY ROUND $r+1 = h'+k'$.

  CHOOSE $h'=h$, $k'=k-1$ $\Rightarrow$ $L_v(h,k^*-1)$ ALREADY SENT.

$$\Rightarrow \text{ONLY } \underbrace{L_v(h,k) \setminus L_v(h,k-1)}_{\text{CONTAINS AT MOST 1 ELEMENT } (k-(k-1))} \text{ NOT SENT YET.}$$

THE ALGORITHM MAKES SURE THAT THE SMALLEST ELEMENT OF L'_v GETS SENT.

REM.I $\Rightarrow$ IF $L_v(h,k) \setminus L_v(h,k-1) \neq \emptyset$ IT IS EXACTLY $\min(L'_v)$

$\Rightarrow$ THIS ELEMENT IS SENT IN ROUND $r+2$

THIS CONCLUDES STATEMENT (2) FOR $r+1$ $\square$

REM: THE BFS-COMPUTATIONS CAN BE DERIVED FROM THE DISTANCES AND ARE PIPELINED VIA MESSAGES $(d(v,s),s)$, WHILE THERE ARE FURTHER MESSAGES (d_s, s) WITH $d_s > d(v,s)$ SENT / DISCARDED.

THIS COMPLETES THE PROOF OF THM . $\square$

DEF: SET $DOM \subseteq V$ IS k-DOMINATING IF FOR ANY $w \in V$, THERE IS A $v \in DOM$ s.t. $d(v,w) \leq k$.

THM: THERE IS A k-DOMINATING SET OF SIZE $O(n/k)$. ON ANY GRAPH G.

PROOF: LET T BE A BFS-TREE OF G. LET $\ell(v)$ BE THE LEVEL OF v IN T. DEFINE $T_i := \{v \in V \mid \ell(v) = i\}$ FOR $i=0,\dots,depth(T)$.

$$D_i := \bigcup_{j \geq 0} T_{i+j(k+1)} \quad \text{FOR } i=0,\dots,k.$$

AS THERE ARE n NODES AND $k+1$ SETS D_i, THERE IS ONE D_i OF SIZE $O(n/k)$. CLEARLY D_i IS k-DOMINATING $\square$

THM: THE FOLLOWING ALGORITHM COMPUTES A k-DOM SET OF SIZE $O(n/k)$ IN TIME $O(D)$

ALGO: $N(v,i) = 0$; (FOR ALL $i \in \{0, \dots, \text{depth}(T)\}$

PARTICIPATE IN COMPUTING T AND $\ell(v)$;

$N(v, \ell(v)) = 1$;

FOR $i = \ell(v), \dots, \text{depth}(T)$ DO

  SEND $N(v,i)$ TO PARENT IN T;

  RECEIVE $N(w, i+1)$ FROM EACH CHILD w IN T;

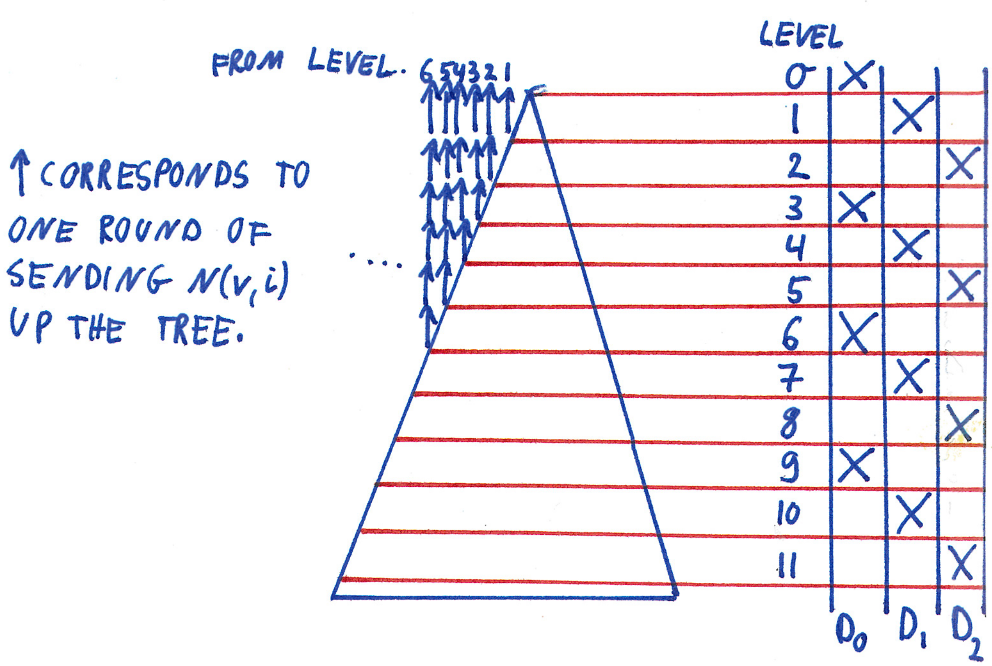
  $$N(v, i+1) := \sum_{w \in T'(v) \setminus \text{PARENT}} N(w, i+1);$$

ROOT NOW COMPUTES $|D_i| := \sum_{j=0}^{\lfloor n/k \rfloor} N(\text{root}, k \cdot j + \ell)$;

ROOT COMPUTES $i' := \operatorname{argmin}_{i \in \{0, \dots, k\}} |D_i|$;

RETURN $D_{i'}$;

EX:



THM: THE FOLLOWING ALGORITHM COMPUTES A $(1+\varepsilon)$-APPROXIMATION OF THE DIAMETER IN $O(\frac{n}{D\varepsilon} + D)$ ROUNDS.

ALGO: COMPUTE $D' := 2 \cdot depth(T)$; // T IS A BFS TREE

$k := \lfloor \varepsilon D'/4 \rfloor$;

COMPUTE k-DOMINATING SET DOM OF SIZE $O(n/k)$;

PERFORM (DOM, D', |DOM|)-DETECTION;

$d_{max} := \max_{\substack{u \in DOM \\ v \in V}} d(u,v)$; // AGGREGATE VIA BFS-TREE

RETURN $(1+\varepsilon) \cdot d_{max}$;

PROOF OF THM: RUNTIME: COMPUTING D', DOM, d_{max} TAKE $O(D)$ TIME EACH. PERFORMING (DOM, D', |DOM|)-DETECTION TAKES $D' + |DOM| + 1 = O(D + n/k)$

$= O(D + \frac{n}{D\varepsilon})$

APPROX-FACTOR: D' IS 2-APPROXIMATION OF D.

1) IT IS $d_{max} \leq D \Rightarrow (1+\varepsilon) d_{max} \leq (1+\varepsilon) D$.

2) IT IS $d_{max} \geq D - k$, AS DOM IS A k-DOMINATING SET. $\Rightarrow (1+\varepsilon) d_{max} \geq (1+\varepsilon) D \cdot (1 - \lfloor \varepsilon D'/4 \rfloor)$

$\geq D$

(1)+(2) YIELD $D \leq (1+\varepsilon) d_{max} \leq (1+\varepsilon) D$.

$\Rightarrow (1+\varepsilon) d_{max}$ IS A $(1+\varepsilon)$-APPROXIMATION OF D.

$\square$