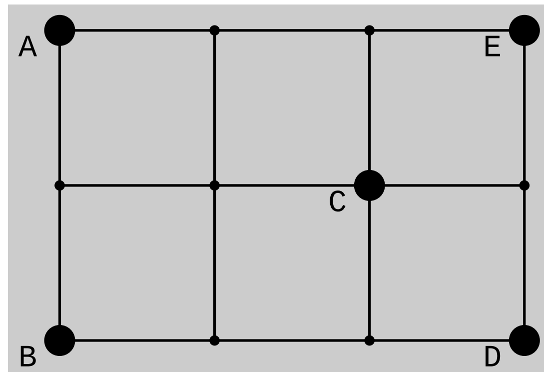
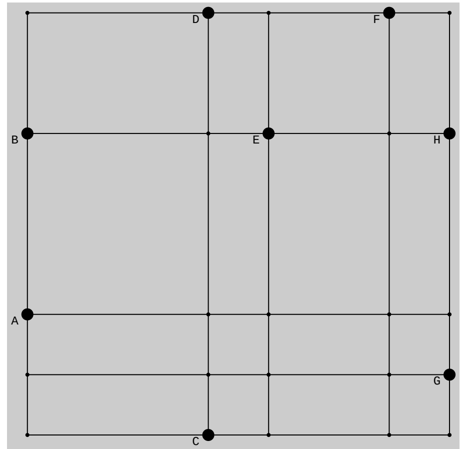
rectilinear steiner tree problem

input: rectilinear grid graph, k terminals



https://webdocs.cs.ualberta.ca/~hayward/304/asn/GanleyC94.pdf

Hanan grid subgrid induced by lines through terminals

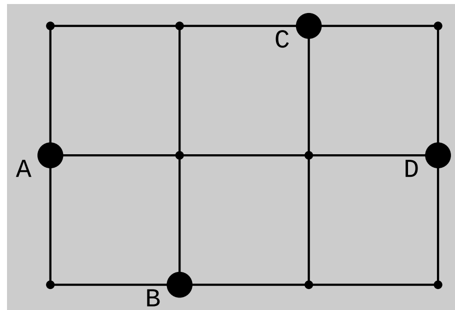


above: Hanan grid of an 8 pins problem on some n×n grid

Theorem: some optimal RST uses only edges of Hanan grid

conclusion: k terminals, effective grid size ≤k×k < n×n

solving small RST problems (Polya *how to solve it*)



bounding rectangle **(brect) smallest containing rectangle**

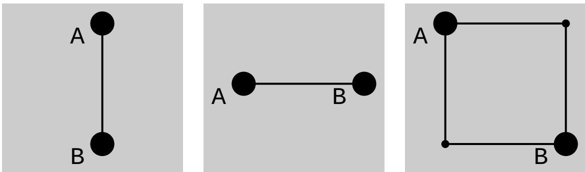
xspan **brect horizontal span (above: 3)**

yspan **brect vertical span (above: 2)**

solving small RST problems

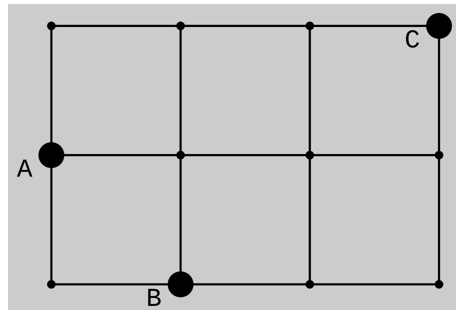
- `|T|=1` **cost 0**

solving small RST problems



- |T|=2   **cost = rectilinear distance = xspan + yspan**

solving small RST problems



- `|T|=3` **cost = xspan + yspan**

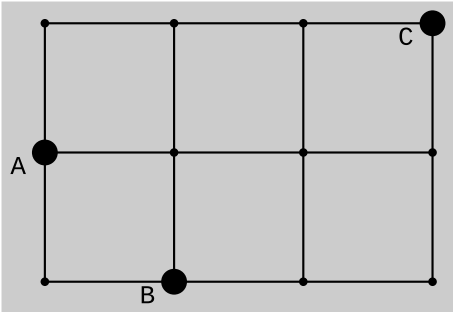
proof: use shrink theorem

shrink theorem: assume left side has !1 pin, say at (0,j)

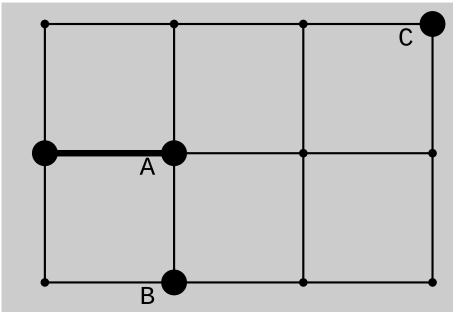
and that min x-coord. of other pins is v.

then some solution tree has path (0,j) to (v,j). (proof omitted)

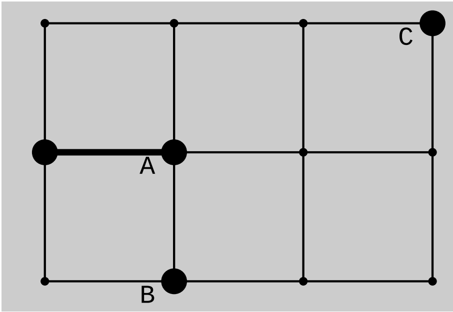
illustrating shrink theorem



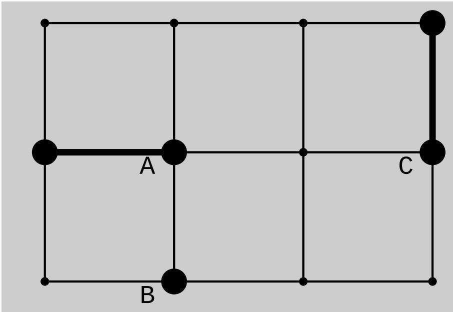
shrink from left: move A from (0,1) to (1,1), add path

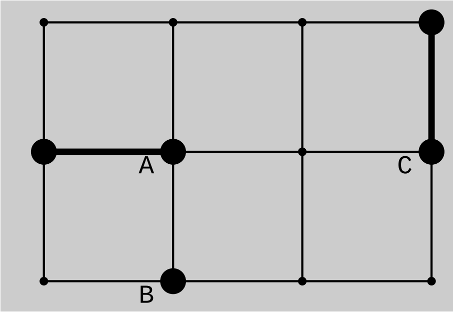


continue on problem A(new), B, C

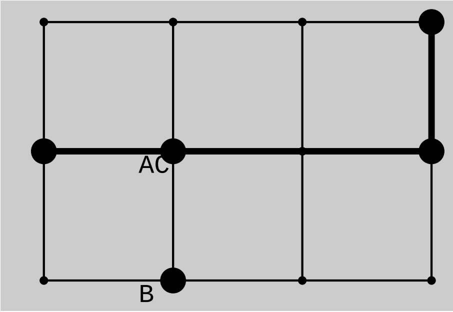


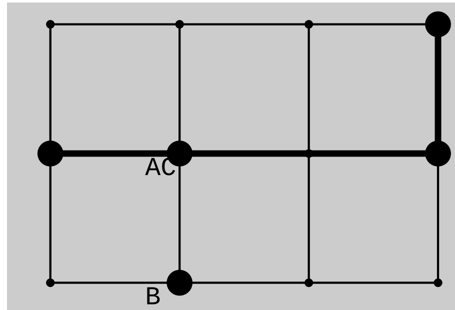
shrink from top: move C from (3, 2) to (3, 1), add path



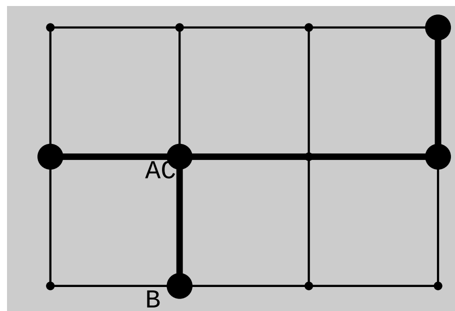


shrink from right: move C from (3, 1) to (1, 1), add path

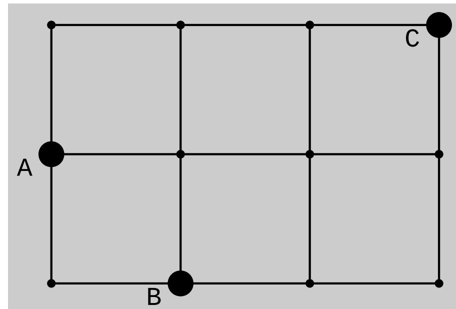




subprojblem has 2 terminals: AC, B **solve it**



solving small RST problems: proof of claim when $|T| = 3$



case 1: brect volume 0 (point/line), done (induction).

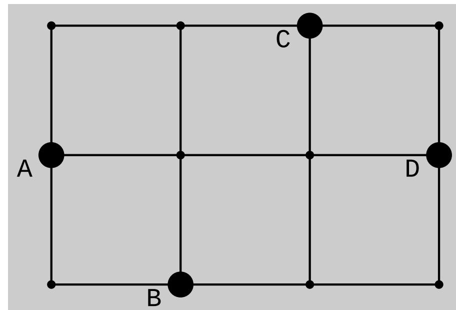
case 2: brect volume positive, 4 sides.

each side ≥ 1 pin (otw shrink brect)

so some side !1 pin (4 sides each $\geq$2 pins then $\geq$4 pins).

Now use the theorem.

solving small RST problems

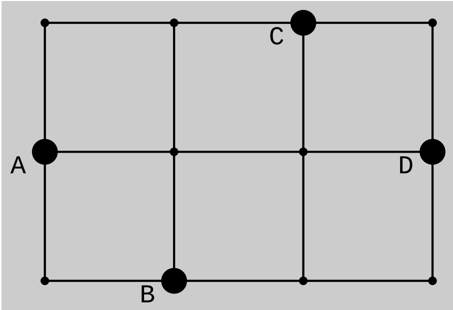


- |T|=4

claim: some side has !1 terminal or each corner has terminal

proof sketch: done unless each side has ≥ 2 terminals, but only 4 terminals, so if this happens must have each terminal in a corner.

solving small RST problems



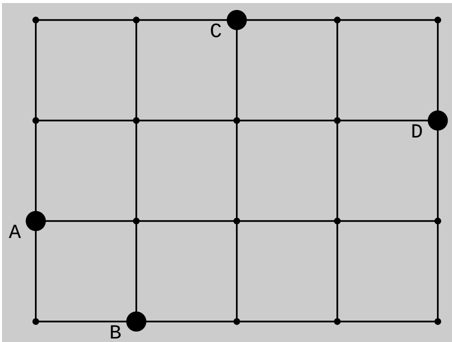
- |T|=4

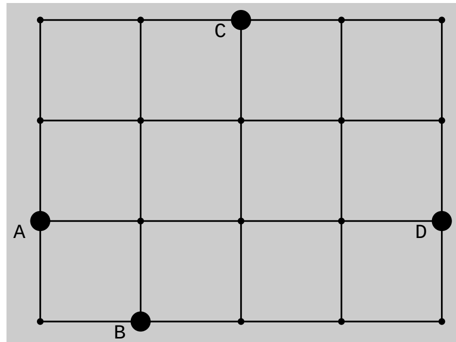
- some side !1 terminal: use shrink theorem
- each corner has terminal:

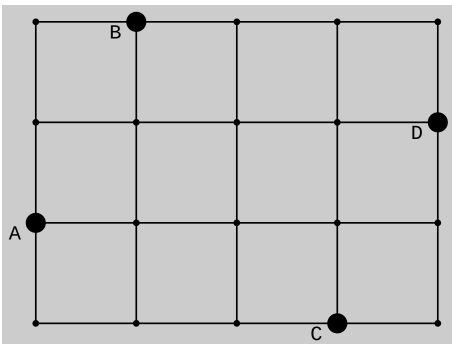
  cost = xspan + yspan + min(xspan, yspan)

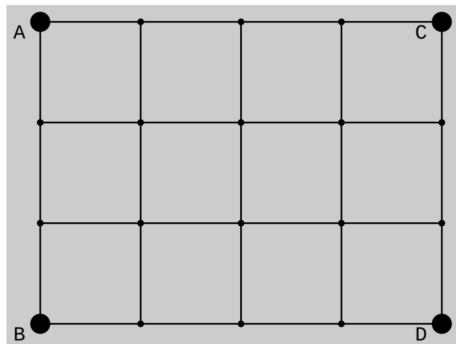
  (proof omitted)

- |T|=4 examples



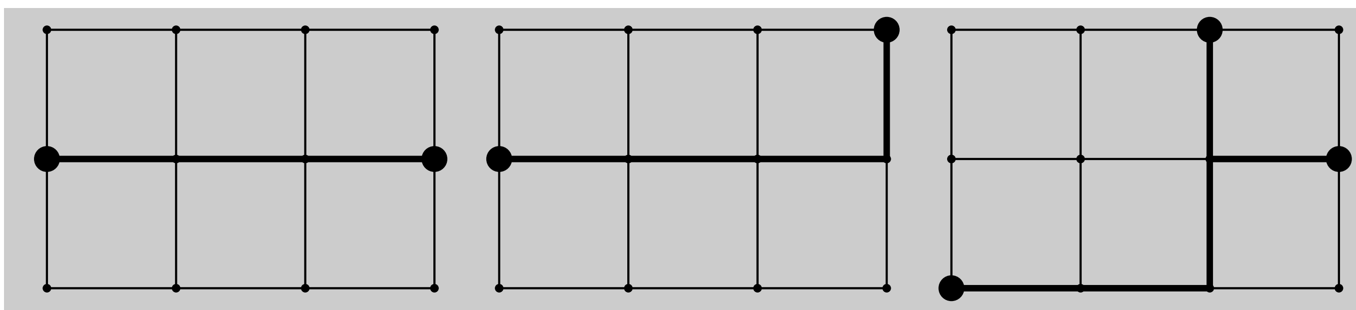




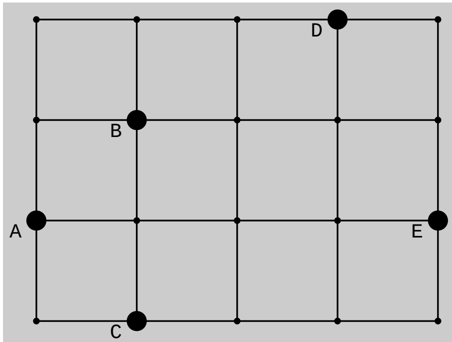


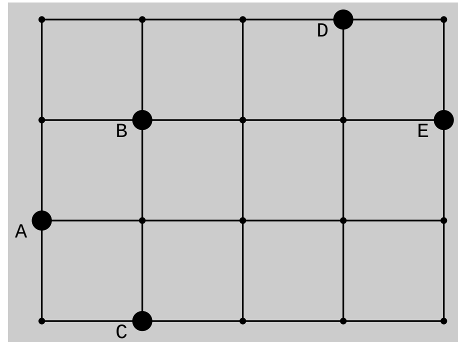
rectilinear caterpillars

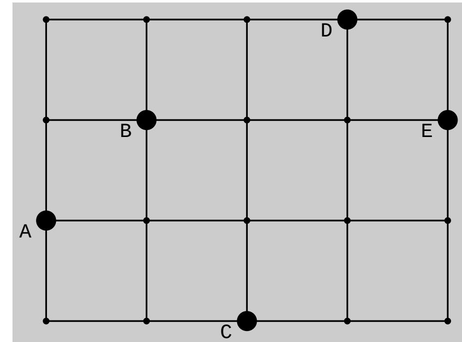
- from side-terminal, horizontal or vertical spine
- each other terminal joins spine (exception: see Ganley et al.)



- three kinds of spines, shown above


D
B
A
E
C


D
B
E
A
C


D
B
E
A
C

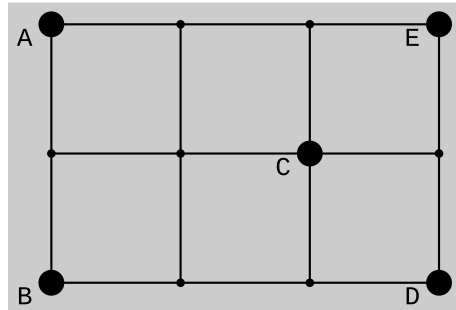
```
alg r-caterpillar(T)
    c ← ∞
    for each side with !1 terminal v:
        for each of 3 kinds of spine:
            d ← cost with spine from v
            c ← min(c, d)
    return c
```

DP decomposition idea

input: set T of terminals

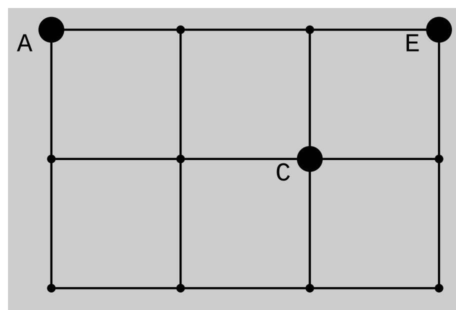
- **for all proper subsets T', solve for T'**
- **assume solve(T) tree has cutpoint C,**

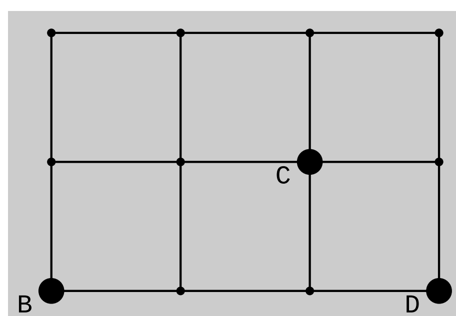
  with T−C subtreee terminal subsets C1, C2
- **combine: return solve($C \cup$ C1) + solve($C \cup$ C2)**

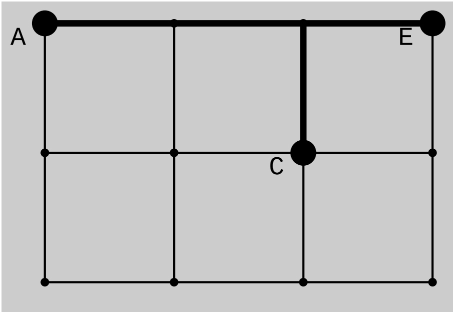


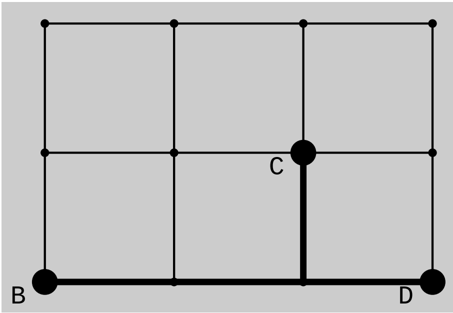
decomposition example: assume solve(T) tree has

cutpoint C, subproblem terminal subsets AEC CBD

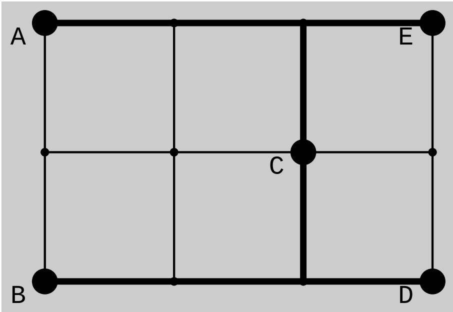


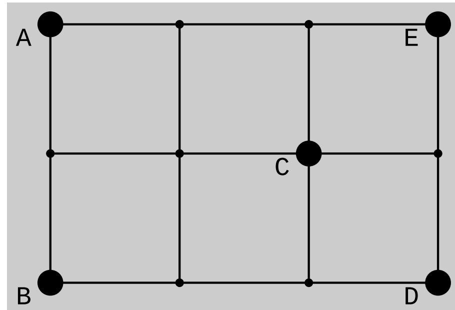






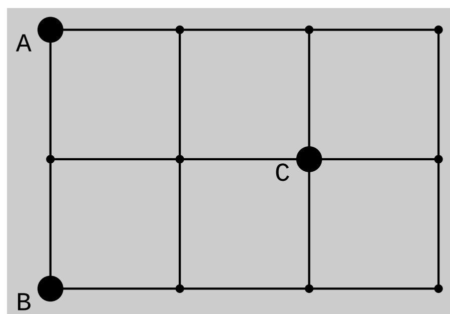
combine solutions

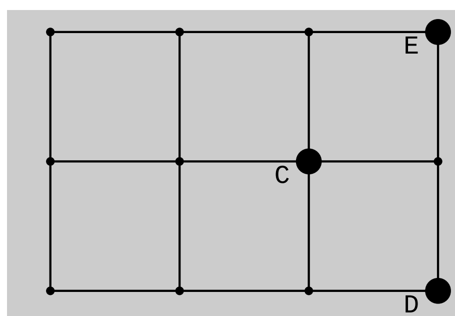


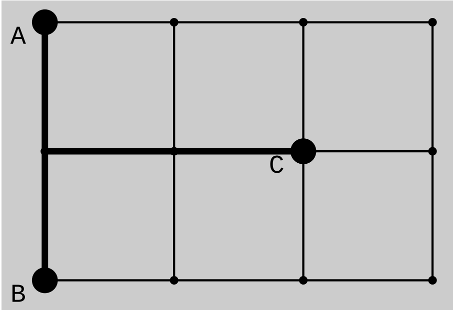


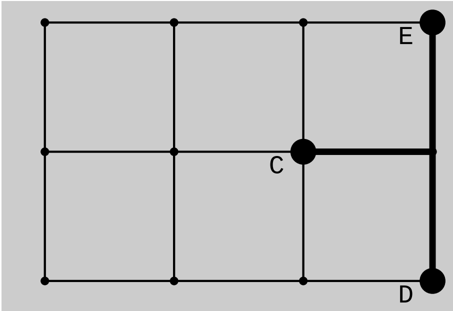
another decomposition example:

cutpoint C, subproblem terminal subsets AEC CBD

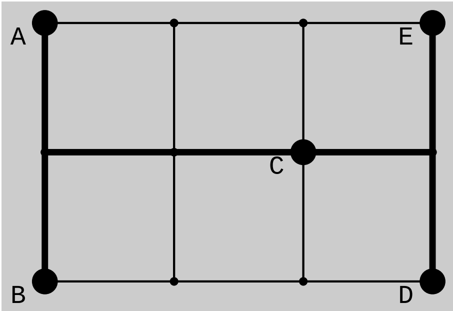








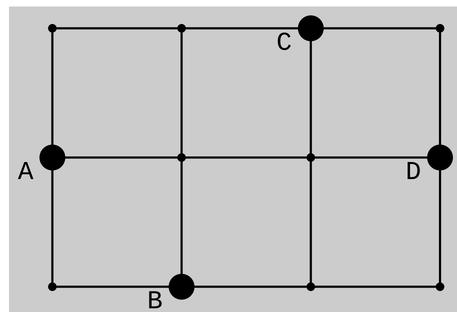
combine solutions



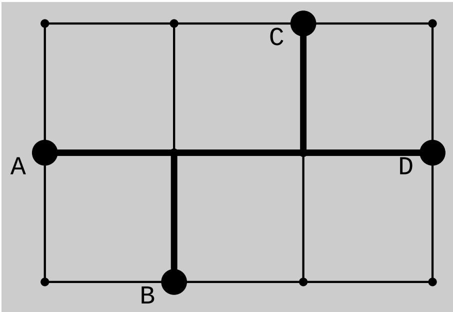
this approach works if, in some optimal solution,

some terminal is a cutpoint

what about this?



only solution



full tree **steiner tree, each terminal is leaf**

full set **steiner tree problem, each optimal tree is full**

theorem: every optimal tree of a full set is r-caterpillar

alg Full Dynamic Program(T)

for all subsets T' of T:

- **$f(T') \leftarrow$ FDP(T')**

f(T) $\leftarrow$ r-caterpillar(T) (best if full set)

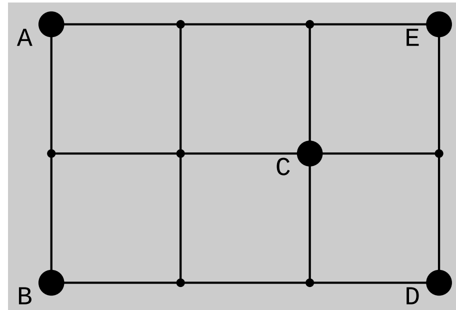
for all possible cutpoints C, all subsets C1,C2:

- **$f(T) \leftarrow \min\{f(T),\quad f(C \cup C1) + f(C \cup C2)\}$**

return f(T)

```
alg FDP(T)   ** Full Dynamic Program **
  if   |T| is 1: return 0
  elif |T| is 2 or 3: return xspan + yspan

  for m <- 2 to |T|:      ** small to large **
    for each subset C of T with |C| == m:
      f[C] <- r-caterpillar(C)  ** best if full set **
      for each j in C:   ** try j as cutpoint **
        k <- 0 if j <> 0 else 1 ** k will always go in Ck **
        for each proper subset S of C \ {j,k}:
          Ck <- {j,k} union S
          Cz <- C \ ({k} union S) ** S proper subset so |Cz| >= 2 **
          f[C] <- min{f[C], f[Ck] + f[Cz]}
```



e.g. j <- A, k <- B

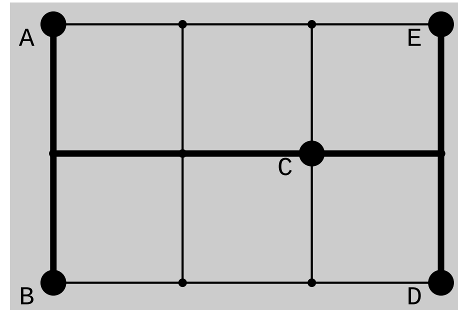
	A	B	C	D	E
A		2	3	5	3
B			3	3	5
C				2	2
D					2

A:	B CDE	2 + 6
	BC DE	4 + 5
	BD CE	5 + 4
	BE CD	5 + 5
	BCD E	6 + 3
	BCE D	6 + 5
	BDE C	7 + 3

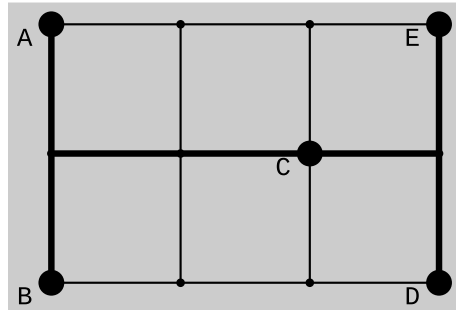
ABC	ABD	ABE	ACD	ACE	ADE
4	5	5	5	4	5

BCD	BCE	BDE	CDE
4	5	5	3

ABCD	ABCE	ABDE	ACDE	BCDE
6	6	7	6	6




```
decomp A B CDE, cost AB + ACDE = 2 + 6 = 8
decomp A BC DE ...
decomp A BD CE ...
decomp A BE CD ...
decomp A BCD E ...
decomp A BCE D ...
decomp A BDE C ...
decomp B ...
decomp C ...
decomp C AB DE, cost ABC + CDE = 4 + 3 = 7
decomp C AD BC, cost ACD + BCE = 5 + 5 = 10
decomp C AE BD, cost ACE + BCD = 4 + 4 = 8
decomp D ...
decomp E ...
```



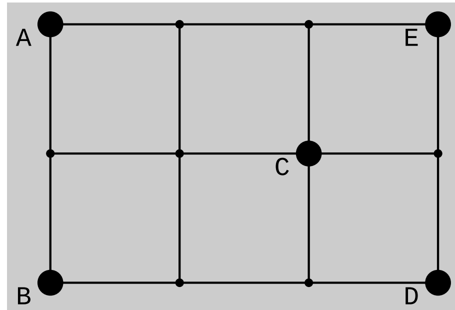
	A	B	C	D	E
A		2	3	5	3
B			3	3	5
C				2	2
D					2

ABC	ABD	ABE	ACD	ACE	ADE
4	5	5	5	4	5
	BCD	BCE	BDE	CDE	
	4	5	5	3	

e.g. j <- A, k <- B

```
A: B CDE   2 + 6
   BC DE   4 + 5
   BD CE   5 + 4
   BE CD   5 + 5
   BCD E   6 + 3
   BCE D   6 + 5
   BDE C   8 + 3
```

best decomp: C AB DE, cost ABC + CDE = 4 + 3 = 7



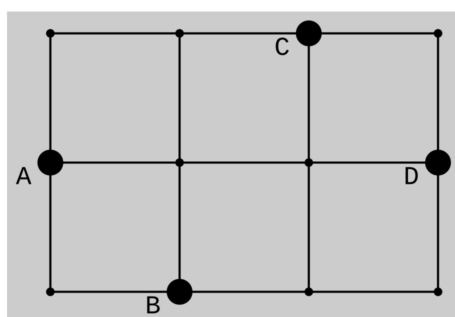
1. if you execute algorithm FDP on this graph, what solution do you find? explain briefly

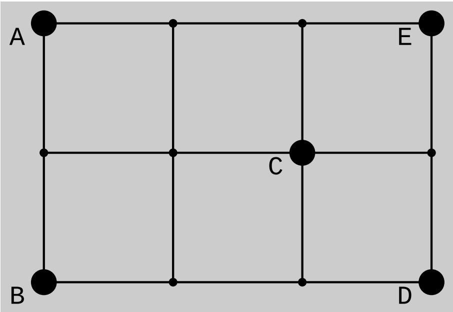
2. repeat the question if you change 2nd line to

`elif T is 2: return xspan + yspan` and

omit line `f[C] <- r-caterpillar(C)`

3. repeat these two questions for the graph below





1. 7. This was traced in these slides.

2. 9. With these changes you will return an MST approx solution.

3. 5. Best solution is a full tree.

4. 7. With these changes you will return an MST approx solution.