europe

icpc.foundation

icpc europe contests

icpc global sponsor programming tools

icpc diamond multi-regional sponsor

icpc europe regional sponsor

icpc cerc sponsor

The 2023 ICPC Central Europe Regional Contest

ICPC CERC 2023

Solution Presentation

University *of Ljubljana*
Faculty *of Computer and Information Science*

Ljubljana, 10. 12. 2023

europe
icpc central europe

E - Equal Schedules (59/60)

Find differences between two schedules.

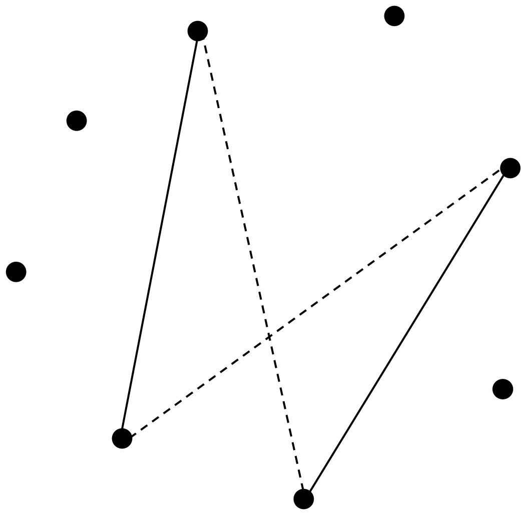
```
0 7 jan            0 9 tomaz           jure -1
7 14 tomaz         9 20 jan            tomaz +1
14 20 jure         20 26 jure
20 24 jan
24 25 tomaz
25 26 jure
```

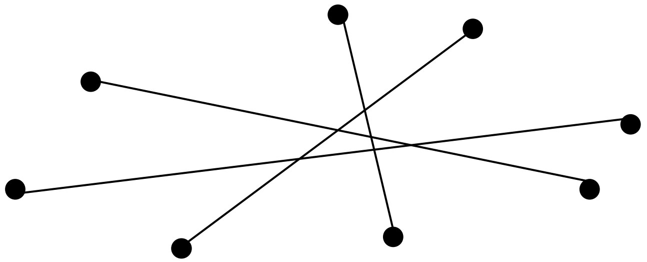
- at most 1000 items
- do what the task says … in any way
  - map/dictionary

B - Ball Passing (46/48)

Pair students arranged in a convex polygon to maximize the sum of distances between paired students.

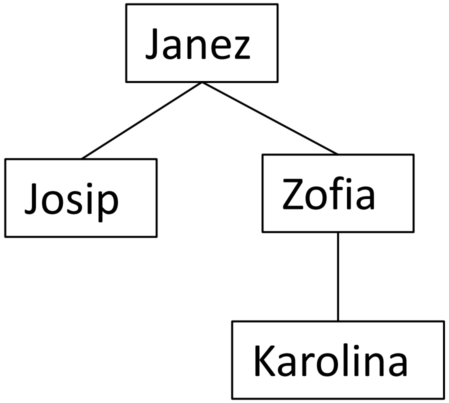
- solve boys and girls separately
- switching non-crossing pairs increases total distance
- unique configuration without non-crossing pairs
  - connect person X with person X+N/2





H - Human Resources (42/46)

Efficiently encode and decode a hierarchical structure.



```
Janez: Josip Zofia
Zofia: Karolina
```

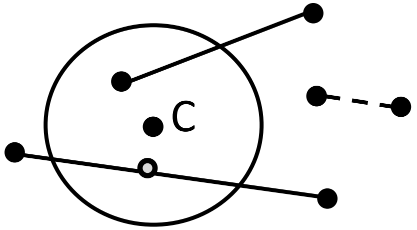
```
Janez
Josip
Zofia
Karolina
(()(()))
```

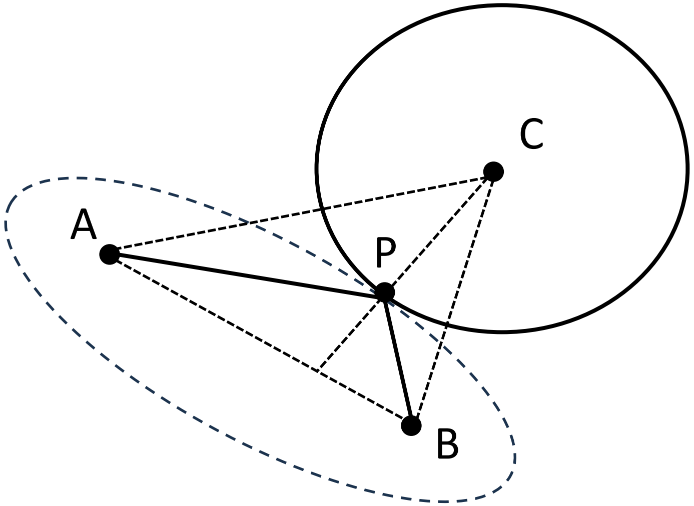
- structure:
  - parent of each node? … n log(n) bits
  - parentheses encoding of a tree … 2n bits
- content:
  - DFS order of employee names
  - preserves preference order of subordinates

G - Going to the Moon (19/35)

Shortest path from A to B while touching disk D = (C, r).

- A or B within D -> straight line
- segment AB crosses/touches D -> straight line
  - C’ = projection of C onto line AB
  - C’ on AB and dist(C’, C) ≤ r
- ellipse with focal points A and B
- search between CA and CB
  - or along AB
  - ternary search, unimodal
  - property: normal at P (CP) bisects ∠APB
- alternatively:
  - sample uniformly around the circle
  - narrow down to the most promising section (and neighbors)



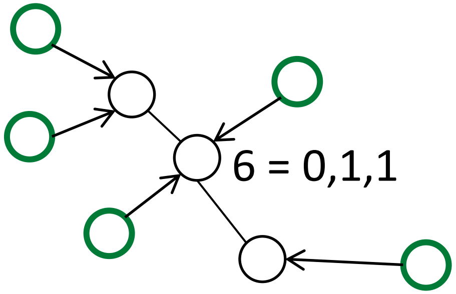


I - Interactive Reconstruction (15/17)

Reconstruct tree structure using sums of neighbors.

- find leaves, determine their parents, cut them off, repeat
- leaves: query 111…1
- parents: query with non-binary values 0 .. n-1?
  - make log(n) binary queries
- too many iterations?
  - adjust answers to previous queries!
  - O(n log n)
- $1 + \lceil \log_2 n \rceil$ queries

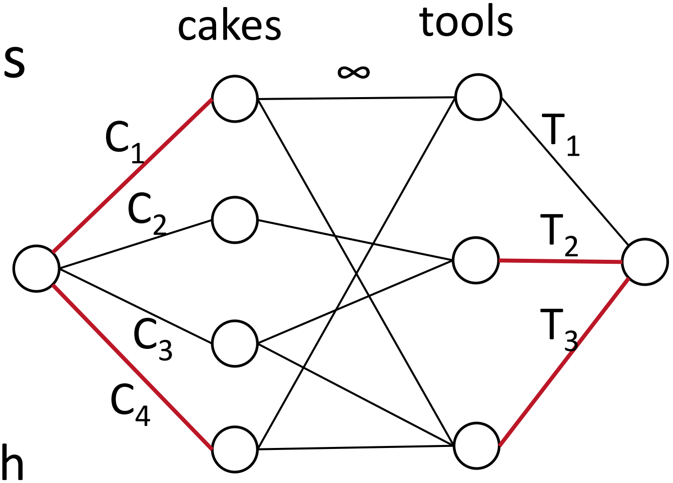
```
01234567
-------------
11111111
01010101
00110011
00001111
```



C – Cakes (11/17)

Decide which cakes to bake to maximize the profit = selling price – production costs (ingredients and tools).

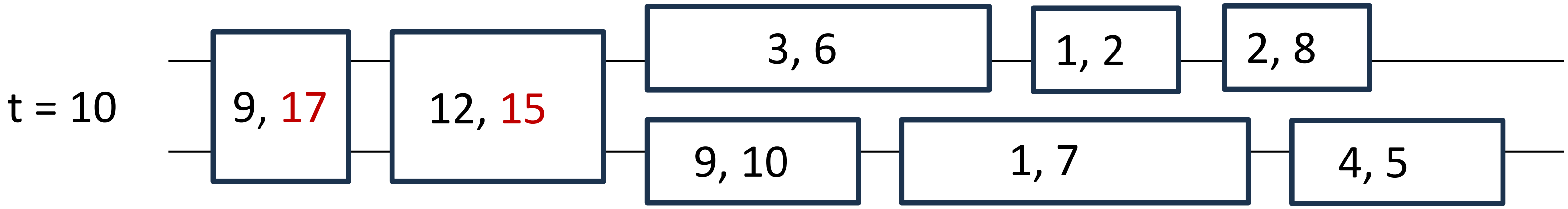
- tools can be reused, subtract ingredients
- model with minimum cut
  - weight = loss of profit
  - cake edge = not selling that cake
  - tool edge = buying that tool
  - min cut = min loss with no source – sink path
  - path = selling a cake without buying a required tool
- large weights … Ford-Fulkerson is too slow
  - capacity scaling, Edmonds-Karp, Dinitz, …



D - Drying Laundry (9/14)

Dry the sheets on two lines of equal length by hanging them over 1 or 2 lines (s=slow time, f=fast time).

- binary search (drying time t)?
  - use a single line if possible (s ≤ t), otherwise hang over two
  - DP r(n, d) … can you reach length d on line 1 with the first n single sheets

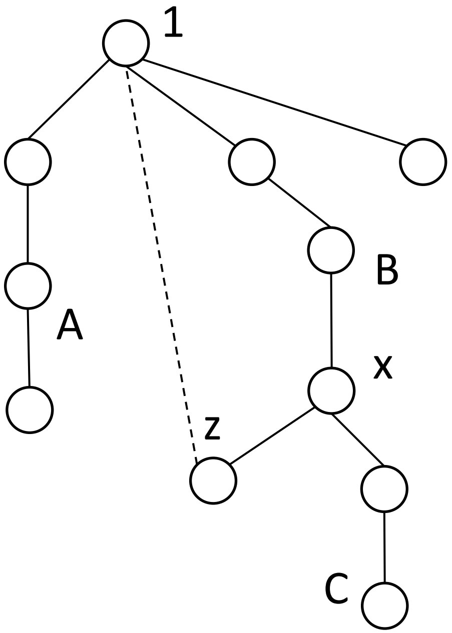


- sort sheets increasingly by their slow drying times
  - increasing t … can afford to hang more sheets on the two individual lines
  - possible? fill one line to capacity … max (d : d ≤ L/2 and r(n, d) = true)
  - answer$_t$ = max(slow time = t, max of fast times for double sheets)
- r(n, *) = OR of two subarrays of r(n-1, *) … bitsets!

K – Keys (4/5)

Arrange an exchange of keys between two residents.

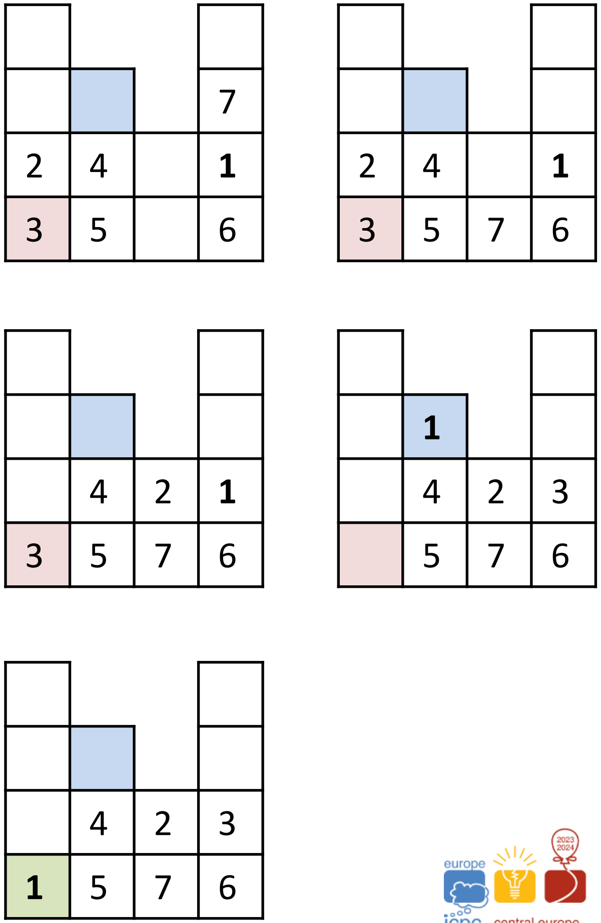
- node 1 (outdoors) not part of any cycle -> no solution
- find such cycle: DFS tree + back-edge (z,1)
- location of node 0 (bedroom)
  - A. in a different subtree
    - Alice: A -> 1 – z (drop A->1) – 1
    - Bob: 1 -> z (grab) -> 1 -> A
  - B. on the cycle
    - Alice: B -> 1
    - Bob: 1 – z -> B
  - C. off the cycle
    - Alice: C -> x (drop C->x) -> 1
    - Bob: 1 – z -> x (grab) -> C



J - Jumbled Stacks (4/6)

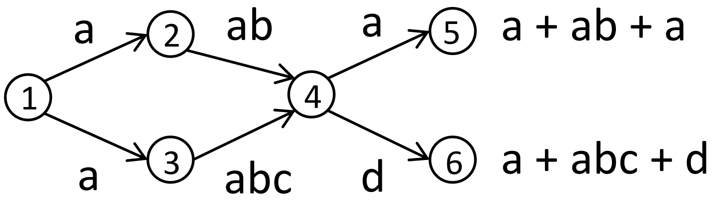
Sort elements in stacks with different capacities.

- you can empty any stack
- move the smallest number x into place
  - from source to target stack
- need a “stage” stack
  - top reserved for x
- clear source stack above x
- clear target stack (avoid source)
  - use top of stage for x to free source stack
- move x into target place
- careful: need at least 3 stacks
  - leave top elements for last



L - Labelled Paths (0/6)

Find lexicog. smallest paths from a single source in a DAG.



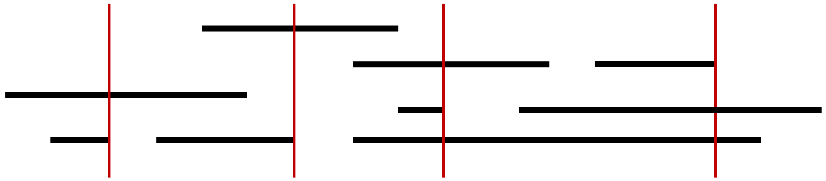
- "Dijkstra" doesn't work
  - track smallest paths of different lengths?
- go backwards from every node
  - reverse topological order, n·m operations
- path = sequence of n substring of superstring A
  - compare: merge with O(n) comparisons of two substrings of A
  - O(n) equality checks and just one comparison!

A	A	B	C	A	B	B	**A**	C	A
A	A	B	C	A	B	B	**B**	A	A

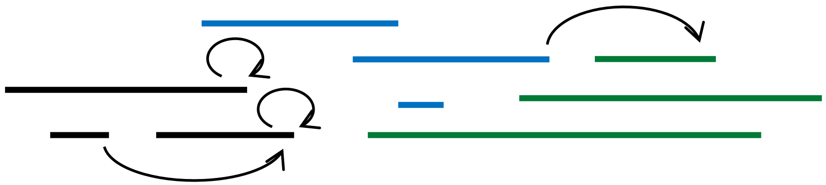
- substring comparison?
  - poly. rolling hash + binary search for longest prefix, O(a + nm(n+log a))
  - suffix array, O(a log a + nm(n+log a))

A – Attendance (0/1)

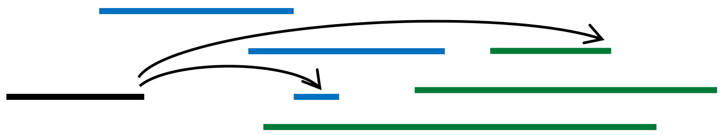
Solve a stabbing set problem on a dynamic set of intervals.



- fixed set of intervals: greedy
  - stab at the earliest end-point
  - sort by ends and solve in O(n)
- dynamic set of intervals
  - simulate the greedy process
  - blocks of size $\sqrt{n}$ (by ends)
  - jump within a block
    - compute the end of the greedy process within a block for every start
  - jump between blocks
  - min end among those with sufficiently large start
  - $O(n\sqrt{n}\log n)$, TLE

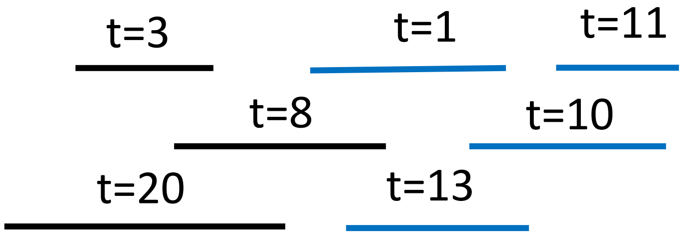


A - Attendance



- exploit known coordinates
  - for every interval precompute
    the “next” (could be inactive) interval within each block
  - enable/disable intervals, update suffix min query data
  - $O(n\sqrt{n})$ time and space, MLE

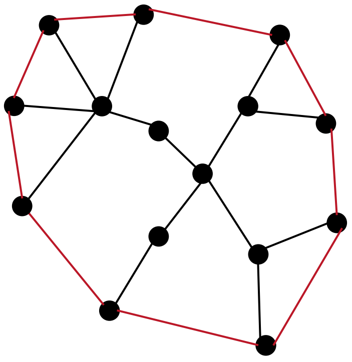
- process queries (different times) “in parallel” from block to block
  - final interval for each “time”
  - sort intervals by starts within a block
  - compute next intervals for current block
    - recompute block on change
    - suffix min data, final positions within block
  - $O(n\sqrt{n})$ time, $O(n)$ space



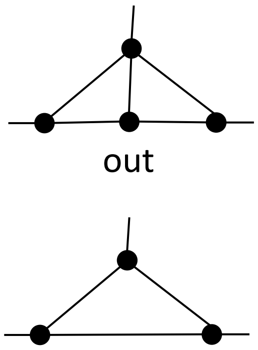
F – Phylogenetics (0/0)

Count colorings of a tree with leaves connected in a circle.

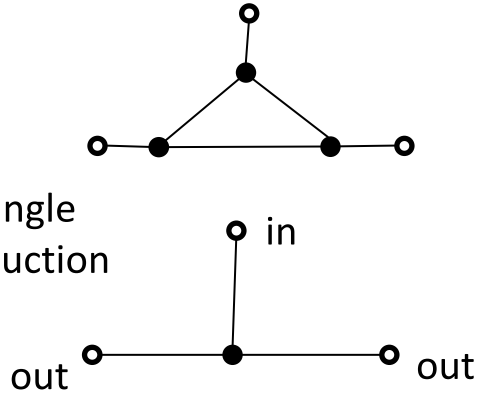
- compress nodes of degree 2 (inner)
- Halin graph
- decompose into a tree and a cycle
  - guess a tree root (node of degree 3 + neighbors)
  - try to compress the graph from leaves towards root
  - triangles exist only on the perimeter
  - linear time O(m), dealing with nodes of degree 3



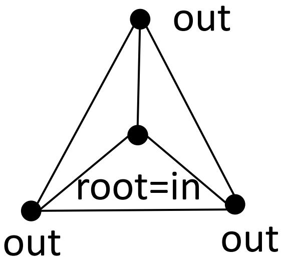
path reduction



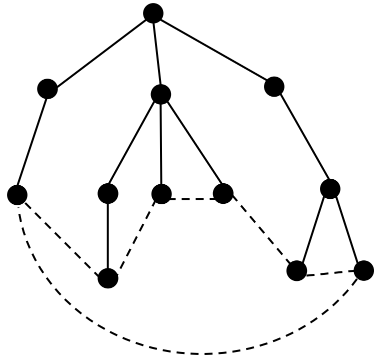
triangle reduction



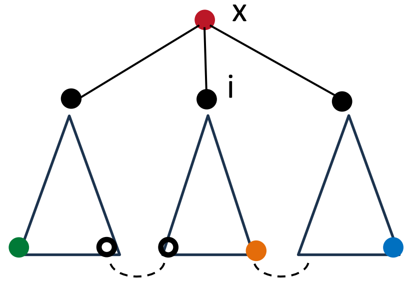
final configuration K4



F - Phylogenetics



- construct a cycle from marked leaves
- reorder children consistently with the cycle
- count colorings with k colors
  - f(x, c, l, r) … x=root, color(x)=c, color(left(x))=l, color(right(x))=r
  - combine subproblems: g(i, d) … first i children end with right leaf of color d
    - choose colors of i, left(i), right(i-1)



- improve to O(n k^4)
  - cumulative sums of subproblems
  - not even close to time limit :(
- exact colors don’t matter, just their equivalence classes!
  - f(x, 3, 5, 3) = f(x, 1, 2, 1)
  - O(n) … O(n) states and O(1) transitions to consider

The End