The task: Given a list of controls with their point values and available time windows, the overall time limit and overtime penalty (points per minute), and the time it takes to travel between each pair of controls (including the start and finish), find which controls to visit and in what order so as to maximize your score.

Establish the problem.

- specifications – task, input, output, legal solution, optimal solution

Task:

The task: Given a list of controls with their point values and available time windows, the overall time limit and overtime penalty (points per minute), and the time it takes to travel between each pair of controls (including the start and finish), find which controls to visit and in what order so as to maximize your score.

It is legal to visit controls more than once or to visit outside the time window, but only one visit (within the window) counts for scoring.

Travel between controls is assumed to always occur at the runner's maximum pace, but it is allowed to stop and wait for any length of time at a control. (It might be advantageous to arrive at a control early and wait for its time window to open.)

- examples

Identify avenues of attack.

- targets
- approach

Series of choices.

- paradigms and patterns

Paradigm: backtracking.

Flavor: ordering (but allow duplicates and omissions)

Pattern: produce output – repeatedly find the next control to visit

(process input is often more awkward for ordering tasks, and is not appropriate here when duplicates and omissions are allowed in the ordering)

– find best solution

- the series of choices

the next control to visit

Define the algorithm.

- size
- generalize / define subproblems
  - partial solution

the sequence of controls visited so far

  - alternatives – for the next control to visit

legal choices: any control (other than the current one) can be visited next, plus the finish

– things to take into account: it is also legal to wait at a control, and it is necessary to be able to show that progress is being made

– making progress: since a complete solution is that the finish has been reached, we have to be progressing towards a case where the finish is the only legal choice for the next control – but it is legal to visit controls more than once and to go overtime… since the goal is the maximum number of points, it is safe to prune any routes where no additional points can be gained → once the latest-ending time window has closed, go directly to the finish (since there's nothing left to gain and points can only be lost)

    - subproblem
- base case(s)

a complete solution – the finish has been reached

- main case

Find the best solution. If the subproblem output is a complete solution:

```
best result so far ← no solution

for each legal alternative for the current choice,
  result = solve the subproblem resulting from choosing that alternative

  if result is a solution and better than the best result so far,
    best result so far ← result

return the best result so far
```

- top level
    - initial subproblem
    - setup
    - wrapup
- special cases
- algorithm

Show termination and correctness.

- termination
    - making progress
    - the end is reached
- correctness
    - establish the base case(s)
    - show the main case
    - final answer

Determine efficiency.

- implementation
- time and space
- room for improvement