

Programming languages – ProLog

Project exercise no. 4 (2020/21)

T. Goluch

1. Implement a `sort(list, sorted)` predicate, that returns a not ascending sorted list. Apply the chosen one sorting algorithm e.g. selection¹, insertion², bubble³, merge⁴, quick⁵ etc ... (**2 pkt.**):
2. Implement `is_graphic(list, response)` predicate, stating whether the list creates a graphic sequence⁶. Use the sort predicate from the previous task. (**3 pkt.**).
3. Implement `is_connected(list, response)` predicate, stating whether the list of vertex degrees creates a graphic sequence from which a connected graph can be created. Use the predicate from the previous task to check whether a graph can be created from vertex degrees list. (**2 pkt.**).

```
: is_connected( [1,0,1], RES )      % disconnected graph
RES = N

: is_connected( [1,1,1], RES )      % non-graphic sequence
RES = N

: is_connected( [1,1,1,1], RES )    % disconnected graph
RES = N

: is_connected( [1,2,2,1,2], RES )  % connected graph
RES = T

: is_connected( [3,3,3,0,3], RES )  % disconnected graph
RES = N
```

In the penultimate case, we can create a path P_5 or cycle/cliue C_3/K_3 and cycle/cliue P_2/K_2 . So, you can create a connected graph.

The last case is an example of an disconnected graph in which the number of edges is greater than the number of vertices.

¹ https://en.wikipedia.org/wiki/Selection_sort

² https://en.wikipedia.org/wiki/Insertion_sort

³ https://en.wikipedia.org/wiki/Bubble_sort

⁴ https://en.wikipedia.org/wiki/Merge_sort

⁵ <https://en.wikipedia.org/wiki/Quicksort>

⁶ <http://mathworld.wolfram.com/GraphicSequence.html>,
<https://mrpandey.github.io/d3graphTheory/unit.html?graphic-sequence>