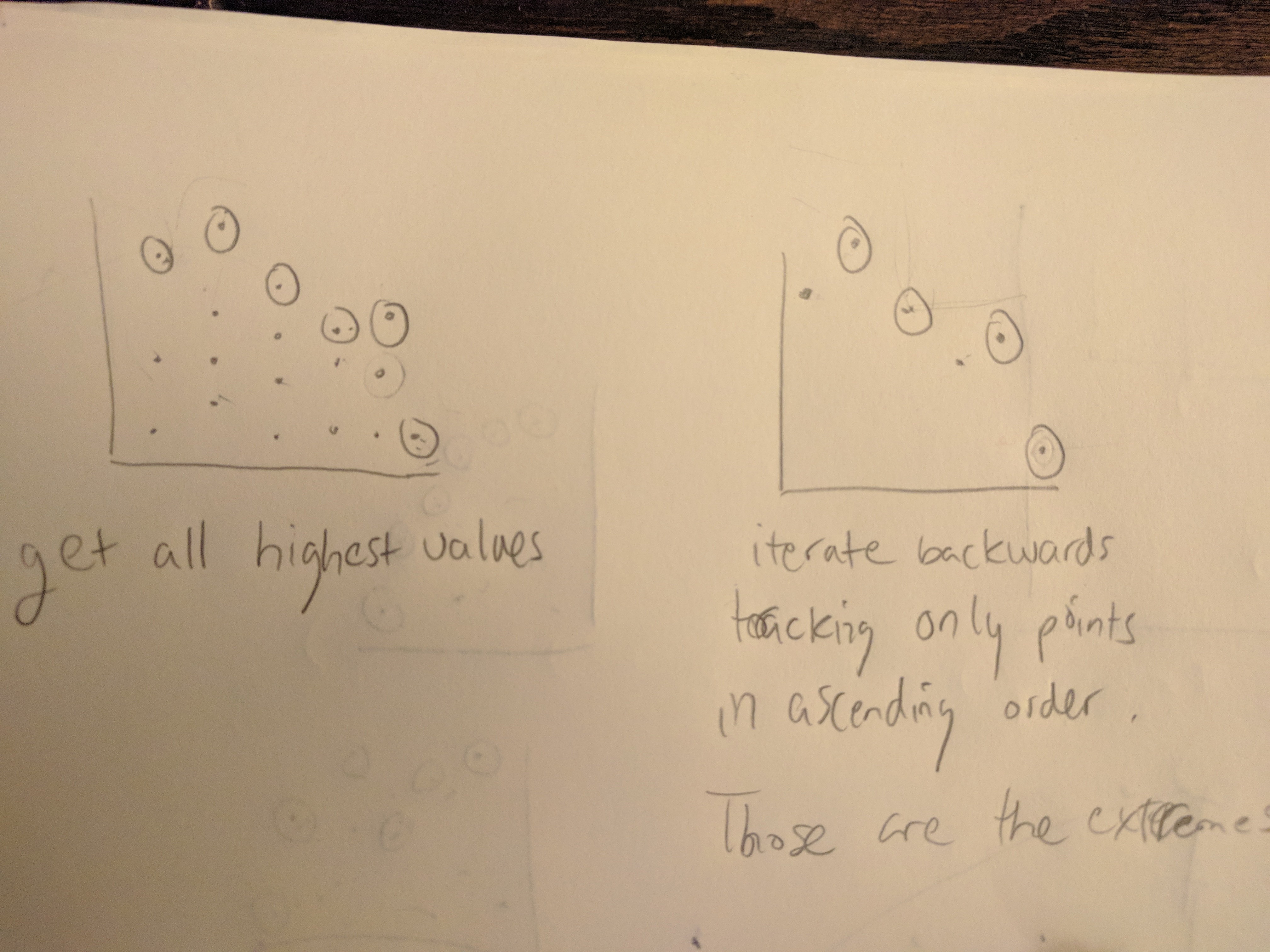
How to run:

* Download file
* Navigate to directory in terminal
* Javac ExtremeFinder.java
* Java ExtremeFinder
* Enter the n value you would like to test

Highlevel Overview:

My algorithm thinks of points in terms of X and Y. It creates an array of linked lists called height that for each value of X stores the point with the larges value of Y. This gives us the fringe. It then goes through these points backwards, making sure to only add points to the extreme group if they are higher than the previous. I tried to show these two main steps below.



|  |  |  |
| --- | --- | --- |
| N | Obvious Algorithm Time | Better Algorithm Time |
| 1000 | 10ms | 1ms |
| 2000 | 22ms | 1ms |
| 5000 | 108ms | 2ms |
| 10000 | 424ms | 3ms |
| 50000 | 10116ms | 10ms |
| 100000 | 41286ms | 16ms |