Marking schedule for A#4 – Monkeys

Total marks = 120 (will be rescaled to 100)

0. **[20 marks] Basic checks**: complies and runs a very simple test, such as abcde (letters only, no spaces).

1. [45 marks] Baseline for sequential mode

1.1. **[5 marks] Validate input** (error panels must be raised if conditions are not met)

1.1.1. [2 marks] PopulationSize must contain a number in the range [1,10000]

1.1.2. [2 marks] TargetText - all characters must be in the set {10, 13, 32-126}

1.1.3. [1 mark] TargetText must contain a text of length in the range [1,300]

1.2. [**30 marks] Simple correctness tests**, with population size 2000 (probably default)

1.2.1. [10=9+1 marks] TC1 correct behaviour and result.

1.2.2. [10=9+1 marks] TC2 correct behaviour and result.

1.2.3. [10=9+1 marks] TC3 correct behaviour and result.

1.2.4. For each case, allocate the last 1/10 mark only if it ends with the green highlight.

1.2.5. No penalty here, but repeat the test and note if generation number not in the [1000, 10000] range (**see section 5 cheating**).

1.3. **[10 marks] Cancel test**: Start again one of TC1-TC3 above and click Cancel button to check whether the evolution stops:

1.3.1. [5 marks] execution stops

1.3.2. [3 marks] no exception

1.3.3. [2 marks] no green highlight (even if it never shows the green highlight).

2. [20 marks] Simple correctness and cancel tests for parallel mode: Run the program in parallel mode, with population size 2000 (probably default)

2.1. **[15 marks] Simple correctness tests**

2.1.1. [5 marks] TC1 correct behaviour and result.

2.1.2. [5 marks] TC2 correct behaviour and result.

2.1.3. [5 marks] TC3 correct behaviour and result.

2.2. **[5 marks] Cancel test**: Start again one of TC1-TC3 above and click Cancel button to check whether the evolution stops:

2.2.1. [2 marks] no exception

2.2.2. [3 marks] no green highlight.

3. **[15 marks] TC4 performance on population size 5000**: on a lab machine or equivalent (4 or 8 logical cores).

   3.1. Run **demo v1** on TC4 five times in parallel and compute the average **demo** parallel real time (you need run this only once per machine/session)

   3.2. Run TC4 three times **sequentially** and compute the average sequential real time.

   3.3. Run TC4 three times in **parallel** and compute the average parallel real time.

   3.4. [5 marks] On typical lab machines, the **parallel** version should run faster than the **sequential** version: at least **20%** faster

   3.5. [5 marks] The **parallel** version should run **5%** faster than **parallel demo v1**

   3.6. [5 marks] The task manager should show high CPU utilisation for the **parallel** version (all logical cores are reasonably busy): the total CPU utilisation box on top left indicates an average at least **40% for 4 cores or 30% for 8 cores**.

   Note that a **sequential** mode cannot achieve much more than 25% on 4 cores or 12.3% on 8 cores (so the above percents seem reasonable)

   3.7. Marks for this section are only allocated if

      3.7.1. the results are correct (with or without highlight)

      3.7.2. both sequential and parallel version end in less than 5 minutes

      3.7.3. there is no algorithm cheating (see section 5)

4. [20 marks] Code inspection

   4.1. [10 marks] The coding differences between the sequential and the parallel modes are well localised; most of the lines should be common between the two modes

      4.1.1. allocate 5 marks only if there is some refactoring but still some duplicates

   4.2. [10 marks] The GUI updates are performed in a thread-safe manner, i.e. a background thread can legally update only

      4.2.1. using SwingUtilities, or

      4.2.2. using explicitly declared (see API) thread-safe controls and methods (such JTextComponent's setText)

      4.2.3. Note that, getText, setEnabled, setBackground are not thread-safe.

      4.2.4. allocate 5 marks only if some GUI operations are thread safe bit not all

5. **[penalty] Cheating code inspection**: cheating the algorithm (not plagiarism)
   5.1. **Must not "cheat" by identifying matching positions**, i.e. must only base the evolution on a global fitness score.
   5.2. You need to check this more carefully if execution is very fast or if the generation number is not in the mentioned [1000, 10000] range.
   5.3. If cheating suspicions, put a flag on all student's marks obtained in **section 3 (performance)** and let us know immediately, for further checks.

6. **[penalty] sequential/parallel code inspection**: in reality providing one mode only
   6.1. **Sequential only**: only marks for **section 1** (i.e. up to **45 marks**); the other section checks can be safely skipped.
   6.2. **Parallel only**: automatically loses section **3.4**, i.e. **5 marks**; maybe others or maybe not.
   6.3. This must be checked by careful code inspection, to see which methods run when you switch the parallel checkbox. If in doubt, you can put a breakpoint or a print statement somewhere in the critical part, to see if the execution reached that point or not
   6.4. **Penalty 2 marks**: no proper population exchange (swap) at the end of the new generation loop

Please ask Radu (r.nicolescu@auckland.ac.nz) if you have questions.

APPENDIX

- **Validation tests** (suggestion)

Population size: -1, 0, 10001

TargetText (suggestion, both longer and with illegal chars):

Погиб поэт! - невольник чести -
Пал, оклеветанный молвой,
С свинцом в груди и жаждой мести,
Поникнув гордой головой*!....*
Не вынесла душа поэта
Позора мелочных обид,
Восстал он против мнений света
Один как прежде.... и убит!
Убит*!...* к чему теперь рыданья,
Пустых похвал ненужный хор,
И жалкий лепет оправданья? -
Судьбы свершился приговор!

- **TC1 on 2000 monkeys**:

There are 10 types of people in this world: those who understand binary and those who don't.

- **TC2 on 2000 monkeys:**

Good friends, good books, and a sleepy conscience: this is the ideal life.

- **TC3 on 2000 monkeys:**

There are only two ways to live your life. One is as though nothing is a miracle. The other is as though everything is a miracle.

TC4 on 5000 monkeys:

To be or not to be, that is the question;

Whether 'tis nobler in the mind to suffer

The slings and arrows of outrageous fortune,

Or to take arms against a sea of troubles,

And by opposing, end them.