Algorithmic Combinatorics | Wi 2016–17
Shagnik Das | Tibor Szabó

# Exercise Sheet 1

**Due date: 14:00, Oct 25th, by the end of the lecture.**
**Late submissions will be erased and written over, à la the Palimpsest.**

You should try to solve all of the exercises below, and submit two solutions to be graded — each problem is worth 10 points. We encourage you to submit in pairs, but please remember to indicate the author of each individual solution.

**Exercise 1** Consider the following algorithm to find the minimum of a set of $2^k$ numbers.

```
Algorithm: MIN
Data: A = {a_1, ..., a_n}, n = 2^k ≥ 2
Result: MIN(A) = min{a_1, ..., a_n}

if n = 2 then
    if a_1 < a_2 then
        return a_1;
    else
        return a_2;
    end
else
    for 1 ≤ i ≤ n/2 do
        set y_i = MIN({a_{2i-1}, a_{2i}});
    end
    return MIN({y_1, ..., y_{n/2}});
end
```

(a) Show that the MIN algorithm requires $n-1$ comparisons to find the minimum element, and that this is the best possible.

(b) After running the MIN algorithm to find the minimal element, how many additional comparisons are required to find the second-smallest element?

(c) Deduce a sorting algorithm that requires $\sim n \log_2 n$ comparisons to sort $n = 2^k$ elements.

**Exercise 2** Consider the following game. I think of an integer $x$ between 1 and $n$, and your job is to try and determine $x$. You are allowed to ask questions of the form "Is $x < y$?" or "Is $x > y$?" for any $y$.

(a) Show that you can find $x$ with only $\lceil \log_2 n \rceil$ questions, and that this is best possible.

To make your job slightly harder, I am now allowed to lie to you at most $k$ times, for some constant $k$.

(b) How many questions do you now need to determine $x$? Provide the best lower and upper bounds that you can find.

**Exercise 3** A man has just bought $n$ horses, and wants to order the horses by speed.[1] However, his private racecourse only has three lanes, so he can only race three of his horses against each other at a time and determine their relative order.[2]

(a) Prove that at least $\lceil \log_6(n!) \rceil$ races are needed to order the horses.

(b) Show that the horses can be ordered within at most $\sim n \log_3 n$ races.

**Bonus (5 pts)** The above bounds are separated by a constant factor. Can you give better bounds to close this gap?

[1]A man has a name, and that name is Ashraf. Our protagonist Ashraf was a maths teacher in Syria, who would occasionally teach a course on algorithmic combinatorics. Having often posed a question on his homework exercises concerning the ranking of horses by speed, he was very excited to finally have the means to put his students' answers to the test.

"Not so fast," you interject, "I know a thing or two about horses. First, they like to eat oats, but," you admit, "that is not very relevant to the discussion at hand. More pertinently, horses are really expensive. How could Ashraf afford to buy $n$ horses?"

You see, Ashraf was an afficionado of classical music, and would play the Oud in his spare time. Advanced in years, he had accumulated a great deal of spare-time practice and mastered the art. However, his talent was only surpassed by his modesty, which kept him from giving any public performances.

This was soon to change, though. One day an entrepreneurial friend of his stopped by his house unannounced, and in doing so heard him practising. Enchanted by the magical music, and sensing the opportunity to make a quick buck, he convinced Ashraf that it was his duty to share his gift with the world. Fast forward a few months, and Ashraf had finished recording his album. After better-than-expected sales in the West, he found himself a rich man, and invested some of the money into the school where he had once taught, then spent the rest on the $n$ horses mentioned in this problem.

"Wait a minute," you exclaim, looking decidedly unconvinced, "I have never heard of the oud, and were it not for Wikipedia, I would not have known that it is a pear-shaped string instrument that, along with the lute, is considered an ancestor of the modern-day guitar. Nor do I expect many of my peers are aware of the oud's existence. How, then, did this album sell so well out here?"

You see, there was a misprint in the album title, which appeared on iTunes as 'Songs of the Ood'. This led several Whovians to purchase the album, expecting to hear telepathic melodies from their favourite show. When they heard Ashraf's instrumental tunes instead, they remained so impressed that they recommended the music to their friends as well. Music historians would, for decades to come, argue whether this misprint was truly an error, or a cunning sales ploy — either way, it was very successful.

[2]He allows sufficient breaks between the races so that the horses do not get tired, and their performance is an accurate reflection of their speed.

**Exercise 4** Given a connected graph $G$ and an arbitrary vertex $v_0 \in V(G)$, show that the breadth-first search starting at $v_0$ returns a tree $T_B$ that preserves distances[3] to $v_0$; that is, for every $v \in V(G)$, $d_G(v_0, v) = d_{T_B}(v_0, v)$.

**Exercise 5** Show that the first $m$ edges added in Kruskal's algorithm form a $m$-edge forest of minimum weight.

**Exercise 6** Somebody suggests the following algorithm for building a minimum weight spanning tree, with the additional feature of always having a connected subgraph throughout the process.

```
Algorithm: LIGHT SPANNER
Data: A connected graph G = ([n], E), ω : E → ℝ
Result: LIGHT SPANNER(G, ω) = T ⊆ E, a minimum weight spanning tree of the
        component of the vertex 1
/* Initialisation:  sort edges, start with empty tree at vertex 1     */
Sort edges by weight, E = {e_1, e_2, ..., e_m}, ω(e_i) ≤ ω(e_j) for all i ≤ j ;
Set C = {1} ;
Set T = ∅ ;
/* Build tree:  always add lightest edge extending component C        */
while true do
    i = 1 ;
    while i ≤ m do // look for first edge extending C
        if |e_i ∩ C| = 1 then // found edge extending C
            T = T ∪ {e_i} ;
            C = C ∪ e_i;
            break;
        end
        i = i + 1;
    end
    if i = m + 1 then // no edge extended C, so C is a connected component
        return T;
    end
end
```

Will this even greedier algorithm succeed? Either prove its correctness or demonstrate

[3] In an unweighted graph $G = (V, E)$, the *distance* $d_G(u, v)$ between vertices $u, v \in V$ is the minimum length of a path from $u$ to $v$. If $u$ and $v$ are in separate connected components, we may take their distance to be infinite.

some input on which it fails.[4]

[4]This may seem a curiously untrusting stance. Indeed, had this suggestion been made in the usual manner — either in a paper, or at a conference, say — then you would no doubt have greeted the new algorithm with joyous enthusiasm, but sadly that was not the case.

Instead, having been deeply impressed by 'Al Gore Rhythms', you decide that you have to take action to save the environment.[5] You manage to secure a meeting with mayor of your city, and pitch your idea of establishing a network of self-driving electric cars. You explain how this will kill three birds with one stone[7]: reducing gasoline-driven cars will curb pollution, lessening demand for gasoline will protect the Earth from drilling and fracking, and autonomous cars will improve road safety.

Impressed, the mayor agrees to put your plan to the test, starting with a trial involving $n$ districts. He asks you how you plan to implement the network while minimising costs. You explain to him that you will use Kruskal's algorithm to find a minimum-cost network, and are halfway through your demonstration of the algorithm when you are taken aback by the unexpected appearance of your rival.

"Yo, buddy," he says, taking advantage of your temporary speechlessness, "I'm really happy for you, I'ma let you finish, but I have one of the best algorithms of all time. One of the best algorithms of all time!" With the obnoxious introduction completed, he goes on to explain how Kruskal's algorithm is "dumb" because it could start by selecting isolated edges, rendering your network useless until it is complete. Given how long it can take to build such a network, he argues that it would be better for the network to grow organically, remaining connected at every step. This way it can be used while it is being built, allowing one to avail of its advantages immediately. To your dismay, the mayor is swayed by his reasoning, and agrees to implement his algorithm provided it will not cost any more than yours.

You sense your opportunity, and are determined to show that the algorithm does not always return a minimum-weight spanning tree, thus heaping on your rival the humiliation he so richly deserves. That said, you would not let your ego trump either your sense of civic duty or your love of mathematics. Hence, if you can prove that the algorithm is in fact correct, you will graciously admit defeat, and will congratulate your rival by sending him a fully-charged Samsung Galaxy Note 7.

[5]You suppose you were most deeply affected by the Vertical Horizon's song about polluting the planet 'Best We Ever Had (Grey Sky Warning)', with the lyrics 'Volkswagens spew toxic fumes/killing all our floral blooms/clouds of dirty black smoke/making pedestrians choke/we cannot keep living like this/somebody do something pls'[6].

[6]Al Gore was not particularly impressed with the use of 'pls' in one of the songs on his album. "What is this word," he asked, incredulously, "and why is it appearing in my album?"

Christoph, his social media manager, explained, "if you want to reach out to the new generation, you have to use textspeak."

"I see," said Gore, begrudgingly, "I'm just afraid that this might hurt my chances of winning the Nobel Prize for Literature."

"k lol," came the unsympathetic reply, and that was the end of that.

[7]While opting for an idiom that is friendlier to the environment.