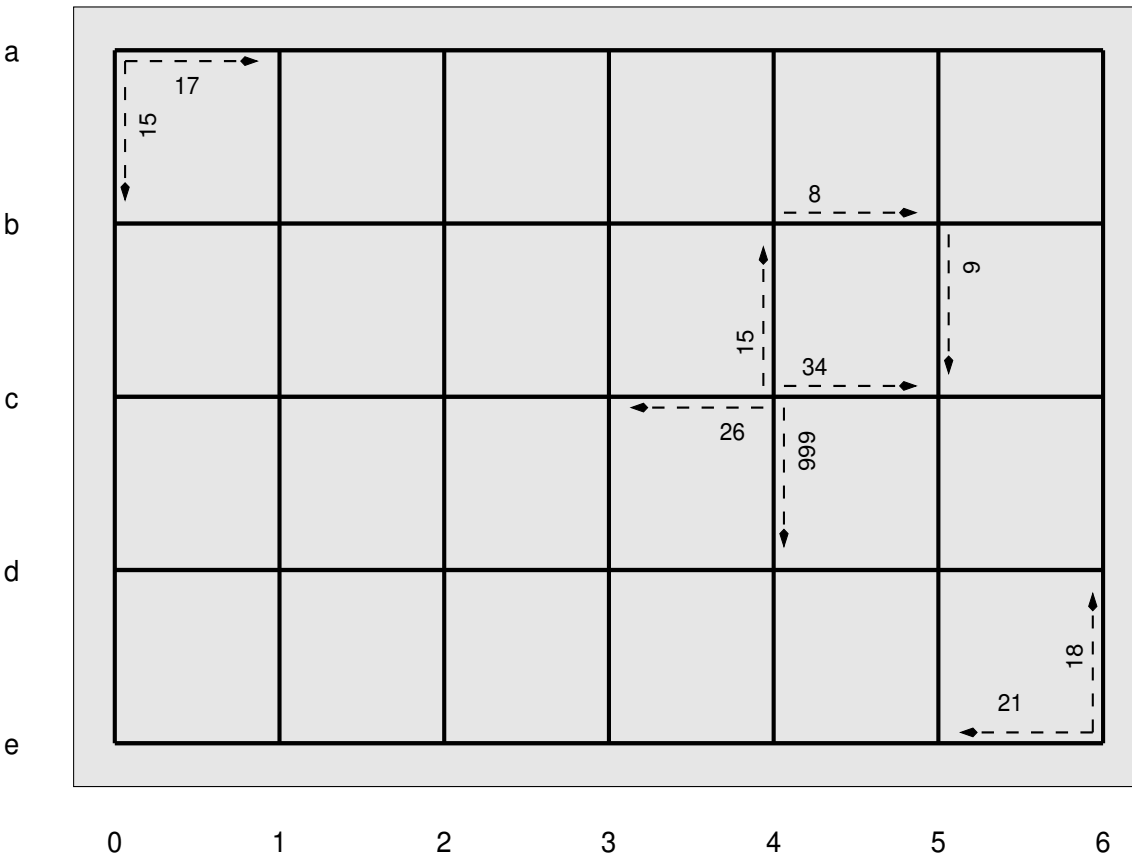


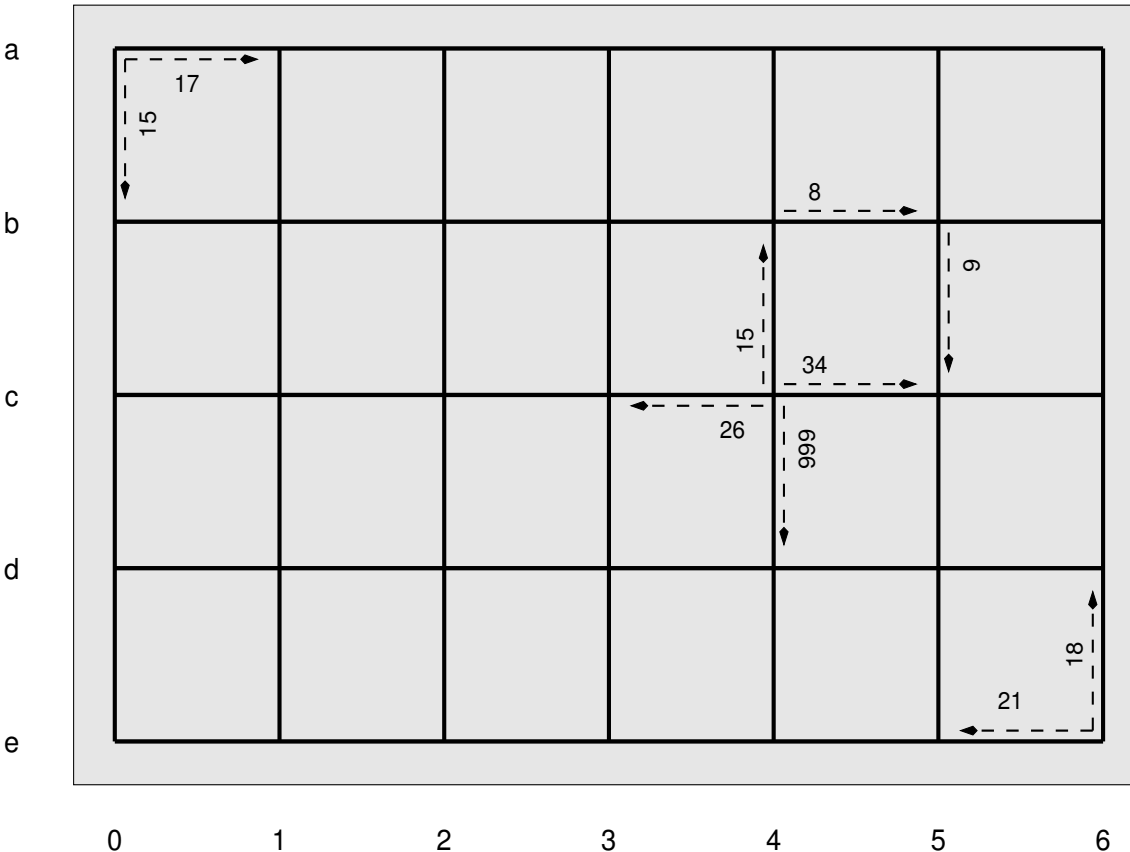
Assignment 2: the grid



The grid defined by:

- number of rows
- number of cols
- array of intersections

Data: the grid & data file



7 5

0a 17 999 999 15

1a ... 999

2a ... 999

...

4c 34 15 26 999

...

4e 999

5e 999

6e 999 18 21 999

4c

5c

1e

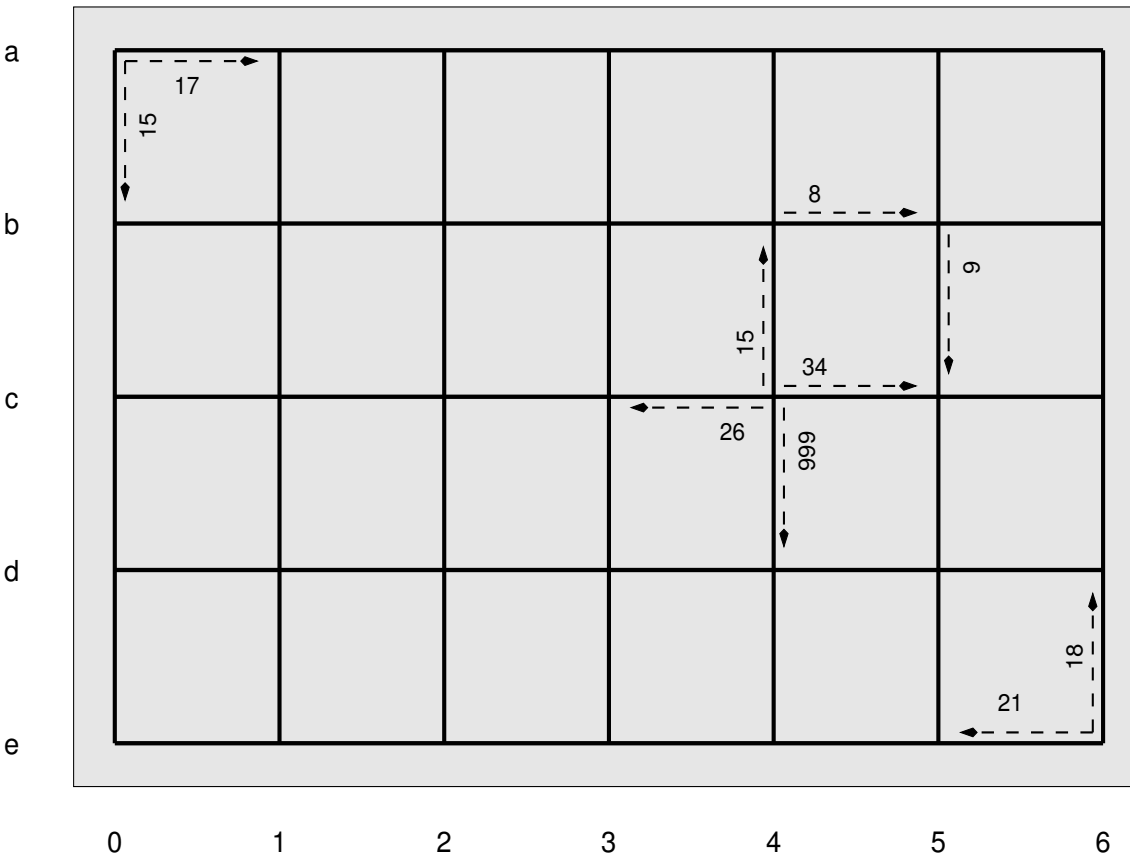
Data: the grid & data file

Data file has 3 parts:

- ncols and nrows
- ncols x nrows intersections, each has 6 data
- some number of locations for stages 2 and 3, each location is a pair col, row

```
7 5
0a  17 999 999  15
1a  ... 999 ... ...
2a  ... 999 ... ...
...
4c  34  15  26 999
...
4e  ... ... ... 999
5e  ... ... ... 999
6e 999  18  21 999
4c
5c
1e
```

Data: the grid



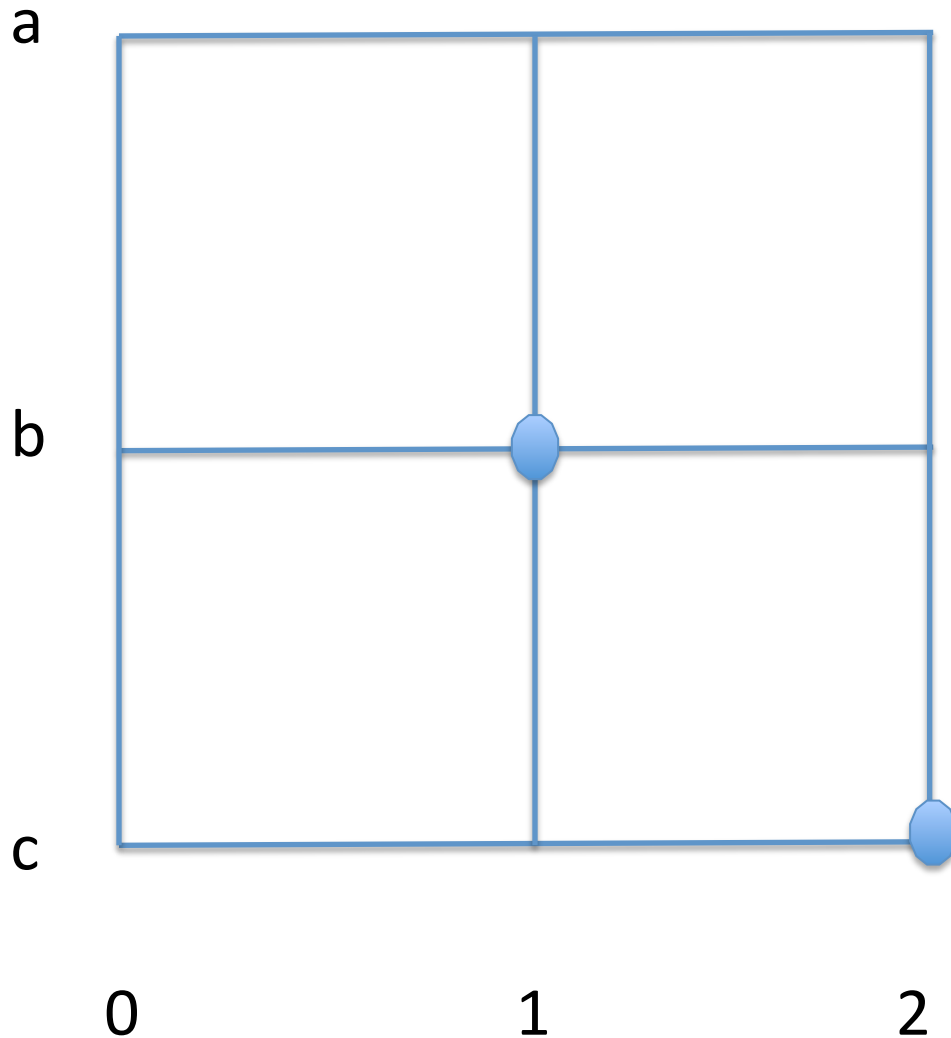
Each intersection has:

- cost for going to each of 4 directions
→ array of 4 int ?

and might have:

- total cost to get here (from somewhere)?
- last move to get here? or preceding intersection?

the grid: example of computing scores



	3	3		
0a	1	999	999	5
1a	1	999	1	1
2a	999	999	3	2
0b	1	1	999	1
1b	6	2	1	3
2b	999	1	1	1
0c	1	3	999	999
1c	1	999	4	999
2c	999	2	999	999
1b				
2c				

QoCT: do it now

QoCT surveys will be open on 2 October 2017 (Week 10).

The link is at:

<https://apps.eng.unimelb.edu.au/casmas/index.php?r=qoct/subjects>

Testing your program

Example testing using `test0.txt` and `test0-out.txt`:

```
$ ./ass2 < test0.txt >mytest0-out.txt  
$ diff mytest0-out.txt test0-out.txt
```

The “`diff`” command will find the difference between 2 files. If it produces no output at all, then the 2 files are absolutely identical (Bravo!). If not, then you need to open both files using `jEdit` and try to figure out what’s wrong in your output.

Assignment 1: CCTS process [SKIP if you know well]

1. **CREATE**: Create a directory, say `ass2`, download all related files into `ass1`, then create `ass2/ass2.c` that satisfies the requirements 😊
2. **COPY**: Copy the whole directory `ass1` to your university's drive `H:`. Note: if you work in lab computers and use `H:`, you don't need to do this step.
3. **TEST**: login into the server `dimefox.eng.unimelb.edu.au`, then on that server, navigate to the directory `ass2`, compile and test your program.
4. **SUBMIT**: while in `dimefox`, submit your `ass2.c`, and verify.

Today Work

Create `ass2.c` for Stage 1, then try all 4 steps. Make sure that you can submit, at least from a lab PC.

Then, incrementally **CREATE** your `ass2.c`, do **COPY-TEST-SUBMIT** after every major development.

1. The CREATE step (on lab PCs or your laptop)

CREATE: Create an assignment's directory, say `ass2`, under your `comp10002`. To this directory:

- download all the data files mentioned in point 2 of FAQ, namely, `test*.txt`, download all `test*.txt` and `test*-out.txt`,
- then create near-empty `ass2/ass2.c`, compile & test to make sure it “works”,
- and implement Stage 1 now.

2. The COPY step (from your laptop)

COPY: Copy the whole directory `ass2` to your university's drive `H:`.

1. If you use your laptop/desktop at home: you need to install `VPN` for remote access to uni's computers. See Alistair's `Submission instructions` from `FAQ` for how to.

2. To copy:

- If yours is a Mac: open a `Terminal`. If it's a PCs: open a `minGW` window [if you don't have `minGW`, install it or alternatively install `pscp` and `putty` as told by `Submission Instructions`]
- Navigate to the parent directory of your `ass1`
- Run the following command for copying the whole directory `ass1`:

```
scp -r ass2 XXX@dimefox.eng.unimelb.edu.au:
```

(note: replace `XXX` with your `loginname`, and don't forget the `colon` at the end of the line; if you use `pscp`, then use that instead of `scp`)

3. The TEST step (supposing that you've ass1.c working)

- **login into the server** `dimefox.eng.unimelb.edu.au`: From Mac **Terminal**, or Windows' **MinGW** window, run command:
`ssh dimefox.eng.unimelb.edu.au`
- **Then**, when you are with `dimefox`:
 - Navigate to your `ass1` directory
 - Compile your program
 - Test, at least with all data Alistair supplied.
- **Example testing** using `test0.txt` and `test0-out.txt`:

```
$ ./ass2 < test0.txt >mytest0-out.txt
```

```
$ diff mytest0-out.txt test0-out.txt
```

The “**diff**” command will find the difference between 2 files. If it produces no output at all, then the 2 files are absolutely identical (Bravo!). If not, then you need to open both files using **jEdit** and try to figure out what's wrong in your output.

4. The submit process

When you are working on `dimefox`, and already navigated to your `ass1` directory, run:

```
submit comp10002 ass2 ass2.c
```

then, wait a few minutes and verify by:

```
verify comp10002 ass2 > my-receipt-ass2.txt  
more my-receipt-ass2.txt
```

The “`more`” command will display the content of the receipt. Alternatively, you can use `jEdit` to open `my-receipt-ass2.txt` for a careful viewing.

When to submit: Submit now, submit today, submit after any session you work with the assignment. Think about submission as a way to backup your work!

Assignments: advices

- *Be active in the subject's Discussion Forum!*
- *Make as many submissions as you want, only the last one (before deadline) counts. Deadline: **10:00AM on Mon 18 September!***
- *To simplify, do submit at uni. If you want to submit from home, then **install VPN today!***
- *Read the specifications carefully.*
- *Test your program carefully, at least with all supplied data. Do the testing not only in your computer, but also on dimefox.*
- ***Read the marking rubric carefully and try to maximize your marks!***
- ***START EARLY, AIM TO FINISH EARLY!***