

CSE 505 HW2 Report Enbo Yu

113094714

14 Oct

Q1:

p cnf n 1

p1 p2 ... pn 0

See Q1.py

Input:

p cnf 3 2

-1 2 0

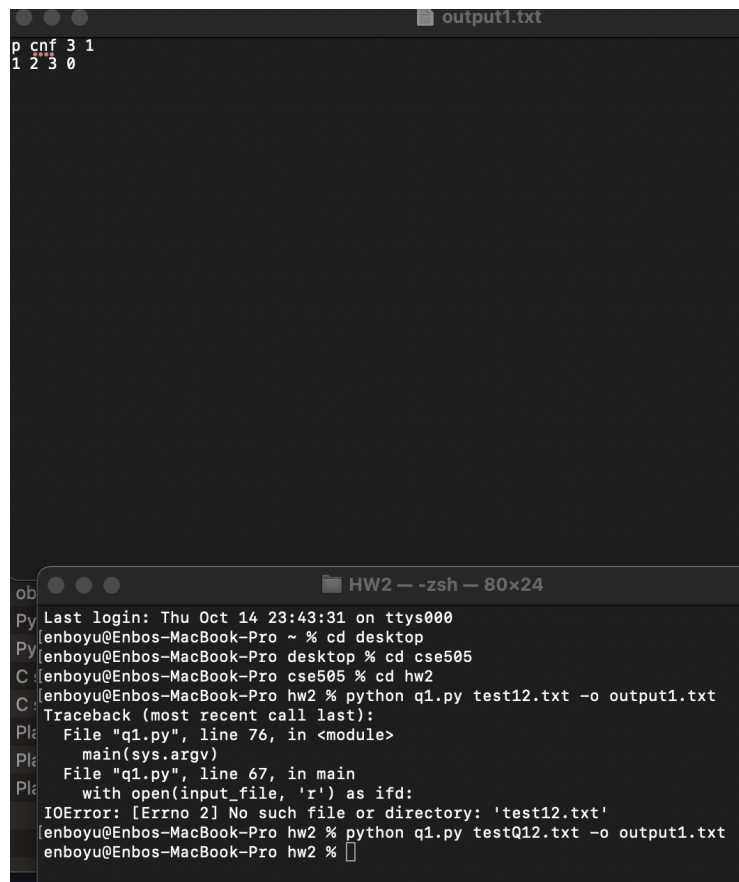
-2 -3 1 0

Output:

p cnf 3 1

1 2 3 0

Q1Test:



```
output1.txt
p cnf 3 1
1 2 3 0

ob
Py Last login: Thu Oct 14 23:43:31 on ttys000
Py [enboyu@Enbos-MacBook-Pro ~ % cd desktop
Py [enboyu@Enbos-MacBook-Pro desktop % cd cse505
C [enboyu@Enbos-MacBook-Pro cse505 % cd hw2
C [enboyu@Enbos-MacBook-Pro hw2 % python q1.py test12.txt -o output1.txt
Traceback (most recent call last):
  File "q1.py", line 76, in <module>
    main(sys.argv)
  File "q1.py", line 67, in main
    with open(input_file, 'r') as ifd:
IOError: [Errno 2] No such file or directory: 'test12.txt'
[enboyu@Enbos-MacBook-Pro hw2 % python q1.py testQ12.txt -o output1.txt
[enboyu@Enbos-MacBook-Pro hw2 % ]
```

Q2:

p cnf n n -p1 0 -p2 0

...- p n 0

See Q2.py

Input:

p cnf 3 2

```
-1 2 0
-2 -3 1 0
Output:
p cnf 3 3
-1 0
-2 0
-3 0
Q2Test:
```

The screenshot shows a terminal window with the following content:

```
p cnf 3 3
-1 0
-2 0
-3 0
```

Below the terminal window, a file explorer view shows the following files and their sizes:

File Name	Size
2 KB Py	2 KB
679 bytes C	679 bytes
701 bytes C	701 bytes
45 bytes Pla	45 bytes
48 bytes Pla	48 bytes
26 bytes Pla	26 bytes

The terminal window also shows the following commands and output:

```
Last login: Thu Oct 14 23:43:31 on ttys000
enboyu@Enbos-MacBook-Pro ~ % cd desktop
enboyu@Enbos-MacBook-Pro desktop % cd cse505
enboyu@Enbos-MacBook-Pro cse505 % cd hw2
enboyu@Enbos-MacBook-Pro hw2 % python q1.py test12.txt -o output1.txt
Traceback (most recent call last):
  File "q1.py", line 76, in <module>
    main(sys.argv)
  File "q1.py", line 67, in main
    with open(input_file, 'r') as ifd:
IOError: [Errno 2] No such file or directory: 'test12.txt'
enboyu@Enbos-MacBook-Pro hw2 % python q1.py testQ12.txt -o output1.txt
enboyu@Enbos-MacBook-Pro hw2 % python q2.py testQ12.txt -o output2.txt
('generated', 'output2.txt')
enboyu@Enbos-MacBook-Pro hw2 %
```

Q3:

```
p cnf n k
p1 p2 ... pk-1 pk 0 p2 p3 ... pk pk+1 0
...pn-k pn-k+1 ... pn-1 pn 0
```

Q3Test:

```
absop@wall:task3$ ./q3.py --help
Usage: ./q3.py <input-file> <k> -o <output-file>
```

Q4:

```
see Q4.c Test: 368 12
23
34
```

45

56

61

25

36

Output: SATISFIABLE

Q5:

1.

For all steps $s \in [0, K]$, integer $\text{num}(s, i, j) \in [0, N \times N - 1]$ denoting which number or label the tile contains at step s .

2.

For all space (i, j) : $\text{num}(0, i, j)$ is the number of the (i, j) in the initial configuration. The empty tile is numbered as 0.

3.

Integers $d_i(s), d_j(s) \in \{-1, 0, 1\}$, respectively denoting the vertical and horizontal movement of the empty tile during the transition from step s to step $s + 1$.

4.

For all steps $s \in [0, K]$, a Boolean $\text{goal}(s)$ denoting whether the puzzle has been solved at step s or at an earlier step.

5.

Integer $\text{empty}_i(s) \in [0, N-1]$ and $\text{empty}_j(s) \in [0, N-1]$, respectively denoting the vertical and horizontal coordinates of the empty tile at step s .

6.

$(\text{num}(s, i, j) = 0) \Leftrightarrow (\text{empty}_i(s) = i \wedge \text{empty}_j(s) = j)$. This synchronizes the two ways of tracking the location of the empty tile.

7.

For all steps $s \in [0, K]$: $(\text{steps} \leq s) \rightarrow (\text{goal}(s) = 1)$

8

For all steps $s \in [0, \text{max}-1]$: $d_i(s) = 0 \vee d_j(s) = 0$. This enforces that the player only makes orthogonal moves.

9.

$\text{goal}(s) = 1 \Leftrightarrow (d_i(s) = 0 \wedge d_j(s) = 0)$. This enforces that no more moves are made once the puzzle has been solved.

10.

$(d_i(s) = 0 \wedge d_j(s) = 0) \rightarrow (\text{empty}_i(s) = \text{empty}_i(s+1) \wedge \text{empty}_j(s) = \text{empty}_j(s+1))$. This enforces that, if one has not moved, the empty tile should not have moved.

11.

For the moves $(i, j) \in \{(1, 0), (-1, 0), (0, 1), (0, -1)\}$: $(d_i(s) = i \wedge d_j(s) = j) \rightarrow (\text{empty}_i(s) + i \geq 0 \wedge \text{empty}_i(s) + i \leq N \wedge \text{empty}_j(s) + j \geq 0 \wedge \text{empty}_j(s) + j \leq N \wedge \text{empty}_i(s) + i = \text{empty}_i(s+1) \wedge \text{empty}_j(s) + j = \text{empty}_j(s+1))$. This enforces that the player only makes valid moves and that the empty tile actually moves.

12.

$\neg((\text{empty}_i(s)=i \wedge \text{empty}_j(s)=j) \vee (\text{empty}_i(s+1)=i \wedge \text{empty}_j(s+1)=j)) \rightarrow (\text{num}(s, i, j) = \text{num}(s+1, i, j)))$. This enforces that, if the tile was not the empty tile during either one of the two steps, its contents should be unchanged.

13.

For all step $s \in [1, K-1]$: $\text{di}(s-1)=1 \rightarrow \text{di}(s) \neq -1$, $\text{di}(s-1)=-1 \rightarrow \text{di}(s)=1$, $\text{dj}(s-1)=1 \rightarrow \text{dj}(s) \neq -1$, $\text{dj}(s-1)=-1 \rightarrow \text{dj}(s)=1$. The number of possible moves at every step reducing the size of the search space.

14.

See Q5.c

Test:

4 100

15 2 1 12

8 5 6 11

4 9 10 7

3 14 13 0

Output: SATISFIABLE

```

eyu@ubuntu: ~/Desktop/cse505hw2
eyu@ubuntu:~/Desktop/cse505hw2$ cd cse505hw2/
eyu@ubuntu:~/Desktop/cse505hw2$ gcc -o Q4 Q4.c picosat.o
eyu@ubuntu:~/Desktop/cse505hw2$ ./Q4
s SATISFIABLE
eyu@ubuntu:~/Desktop/cse505hw2$ gcc -o Q5 Q5.c picosat.o
eyu@ubuntu:~/Desktop/cse505hw2$ ./Q5
s SATISFIABLE
eyu@ubuntu:~/Desktop/cse505hw2$

```