CS130B – Data Structures And Algorithms II

Discussion Section Week 8

Written Assignment 4

Optional

Program 4

Due 11:59 PM, Wednesday June 7

- Implement a randomized algorithm to fit a set of sample points using a line
- Output the coefficients describing the best fit line
- Best fit line has lowest median error across all trials
  - ✘ You may get different best fit lines on different program executions

One More DP Example

Frog Crossing River Problem:

A river is N meters wide. Every meter from the edge there may be a stone or not, indicated by a boolean value in an array S. A frog needs to cross the river. However, if the previous jump was x meters, then the next jump can only be within x-1 to x+1 meters away. Assuming the first jump is 1 meter long, determine whether or not the frog can cross the river.

One More DP Example

Frog Crossing River Problem:

Recurrence relation

$B[x][y] =$ True if frog is at x meters away from the edge and its last jump was y meters

$$= \begin{cases} \text{true if } B[x-y][y] \text{ or} \\ \qquad B[x-y][y-1] \text{ or} \\ \qquad B[x-y][y+1] \\ \text{false otherwise} \end{cases}$$

Example Reduction: 3-SAT to HC

3-SAT is NP-Complete

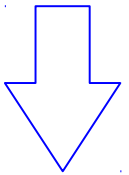
- To show Hamiltonian Cycle (HC) is NP-Complete
  1. Show HC is in P
  2. Show HC is NP-Hard
     - Can be done by taking an NP-Hard problem A e.g. 3-SAT and providing a polynomial time f reduction to B e.g. HC

$$x \in A \leftrightarrow f(x) \in B$$

Reduction: 3-SAT to HC

3-SAT Instance: $n = 3$ variables, $k = 2$ clauses

$$(v_1 \vee v_2 \vee \overline{v_3}) \wedge (\overline{v_1} \vee v_2 \vee v_3)$$



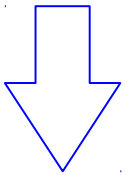
Hamiltonian Cycle Instance:

Construct n paths where path P_i consists of vertices $v_{i,1}, \ldots, v_{i,b}$, where b is larger than k e.g. $b = 3k + k$. There are directed edge $(v_{i,j}, v_{i,j+1})$ and $(v_{i,j+1}, v_{i,j})$.

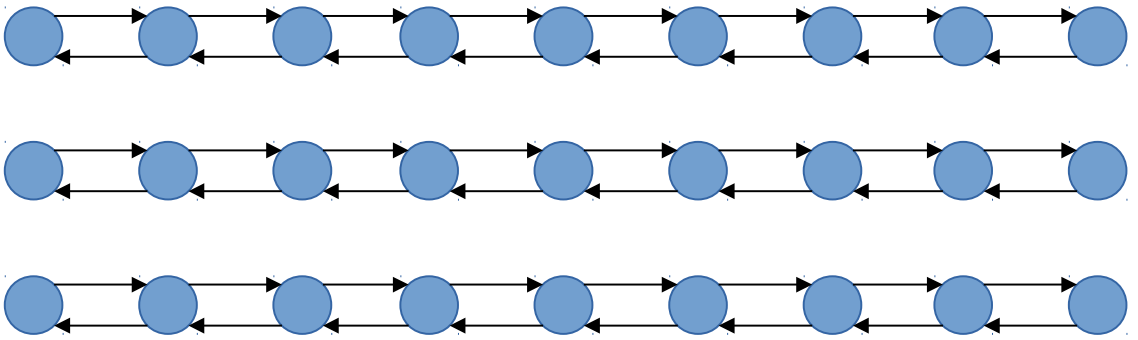
Reduction: 3-SAT to HC

3-SAT Instance: $n = 3$ variables, $k = 2$ clauses

$$(v_1 \vee v_2 \vee \overline{v_3}) \wedge (\overline{v_1} \vee v_2 \vee v_3)$$



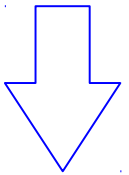
Hamiltonian Cycle Instance:



Reduction: 3-SAT to HC

3-SAT Instance: $n = 3$ variables, $k = 2$ clauses

$$(v_1 \vee v_2 \vee \overline{v_3}) \wedge (\overline{v_1} \vee v_2 \vee v_3)$$



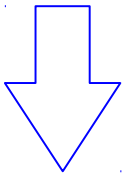
Hamiltonian Cycle Instance:

Add directed edges $(v_{i,1}, v_{i+1,1})$, $(v_{i,1}, v_{i+1,b})$, $(v_{i,b}, v_{i+1,1})$, $(v_{i,b}, v_{i+1,b})$.

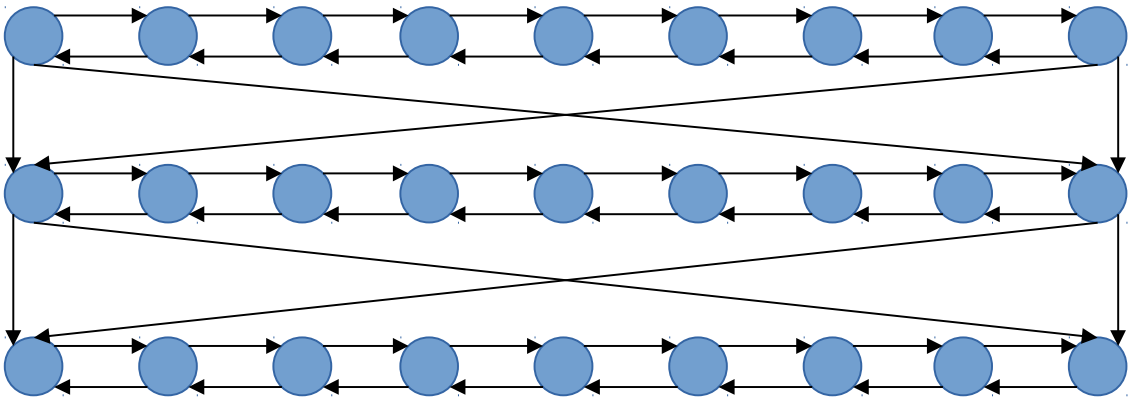
Reduction: 3-SAT to HC

3-SAT Instance: $n = 3$ variables, $k = 2$ clauses

$$(v_1 \vee v_2 \vee \overline{v_3}) \wedge (\overline{v_1} \vee v_2 \vee v_3)$$



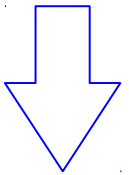
Hamiltonian Cycle Instance:



Reduction: 3-SAT to HC

3-SAT Instance: $n = 3$ variables, $k = 2$ clauses

$$(v_1 \vee v_2 \vee \overline{v_3}) \wedge (\overline{v_1} \vee v_2 \vee v_3)$$



Hamiltonian Cycle Instance:

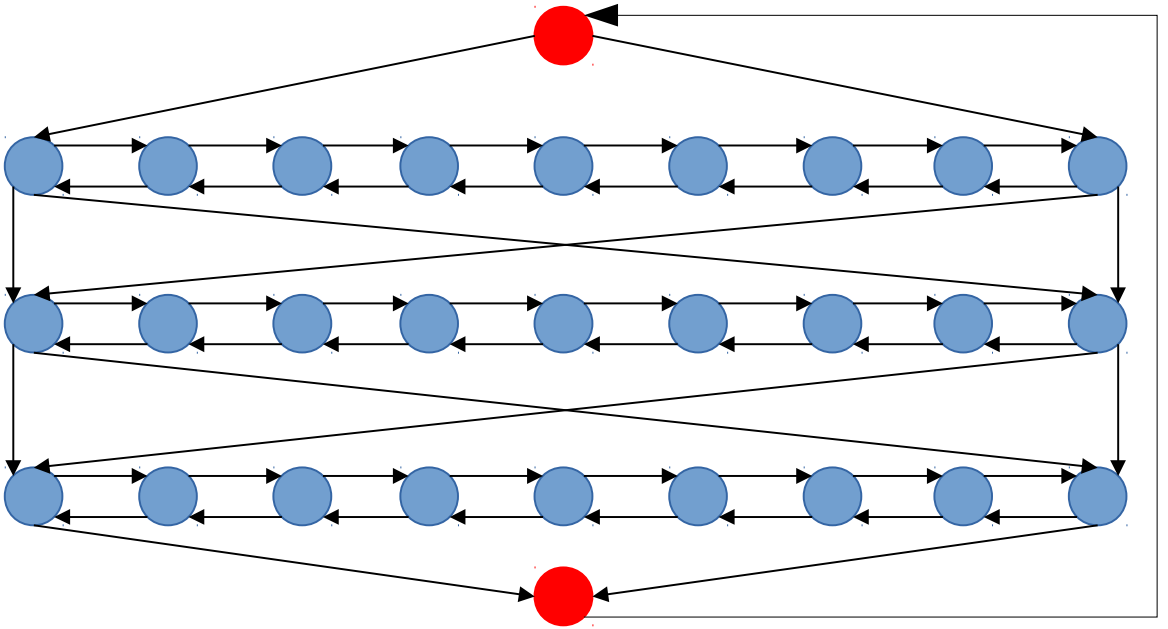
Add vertices s, t and add directed edges $(s, v_{1,1})$ $(s, v_{1,b})$, $(v_{n,1}, t)$, $(v_{n,b}, t)$.

Reduction: 3-SAT to HC

3-SAT Instance: $n = 3$ variables, $k = 2$ clauses

$$(v_1 \vee v_2 \vee \overline{v_3}) \wedge (\overline{v_1} \vee v_2 \vee v_3)$$

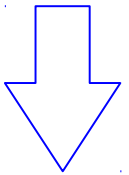
Hamiltonian Cycle Instance:



Reduction: 3-SAT to HC

3-SAT Instance: $n = 3$ variables, $k = 2$ clauses

$$(v_1 \vee v_2 \vee \overline{v_3}) \wedge (\overline{v_1} \vee v_2 \vee v_3)$$



Hamiltonian Cycle Instance:

Left-to-right or right-to-left path through $v_{i,1}, \ldots, v_{i,b}$ corresponds to true or false assignment of variable v_i. Finally, add clause vertices v_{c_j}. If v_i appears in c_j, then add edges $(v_{i,3(j-1)+2}, v_{c_j})$, $(v_{c_j}, v_{i,3(j-1)+3})$; otherwise if $\overline{x_i}$ appears in c_j, then add $(v_{i,3(j-1)+3}, v_{c_j})$,$(v_{c_j}, v_{i,3(j-1)+2})$.

Reduction: 3-SAT to HC

Hamiltonian Cycle Instance: