Homework 4

Mengxiang Jiang
CSEN 5336 Analysis of Algorithms

December 1, 2022

Problem 1. Find the prefix function π for the pattern $P =$ "tortoise" for the Knuth-Morris-Pratt Algorithm. Find out the status q at different steps while searching the pattern in the text $T =$ "distortion tortoise tortilla".

Prefix function $\pi[1 \dots m]$:

i	1	2	3	4	5	6	7	8
$p[i]$	t	o	r	t	o	i	s	e
$\pi[i]$	0	0	0	1	2	0	0	0

Current state values:

From $i = 1 \dots 19$:

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
$T[i]$	d	i	s	t	o	r	t	i	o	n		t	o	r	t	o	i	s	e
$q[i]$	0	0	0	1	2	3	4	$\cancel{1}$ 0	0	0	0	1	2	3	4	5	6	7	$\cancel{8}$ 0

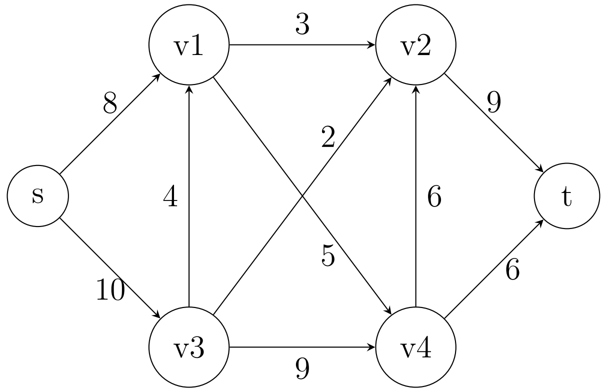
From $i = 20 \dots 28$:

i	20	21	22	23	24	25	26	27	28
$T[i]$		t	o	r	t	i	l	l	a
$q[i]$	0	1	2	3	4	$\cancel{1}$ 0	0	0	0

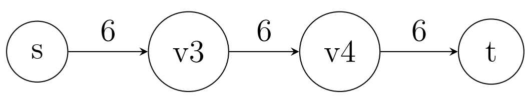
one valid shift at $i = 19 \rightarrow$ shift of $i - m = 19 - 8 = 11$.

Problem 2. Find the maximum flow in the flow network shown in figure 1. In the flow network s is the source vertex and t is the destination vertex. The capacity of each of the edges are given in the figure.

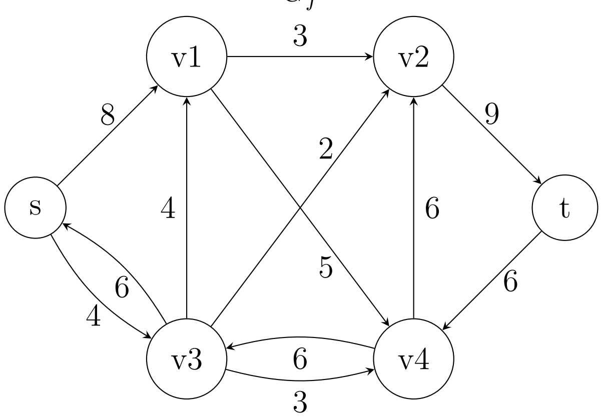
Figure 1: Flow Network



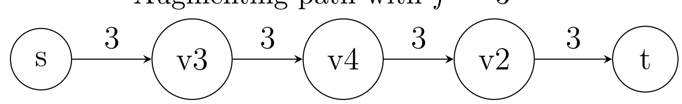
Augmenting path with $f' = 6$



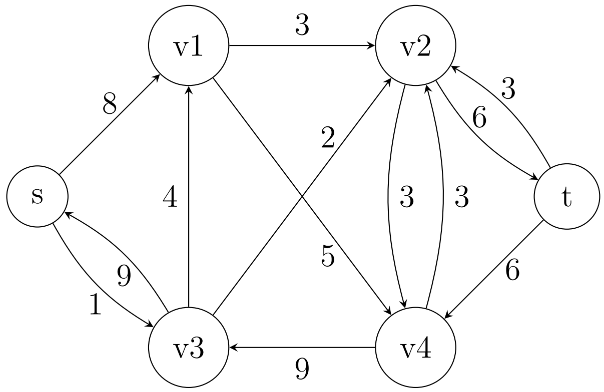
G_f



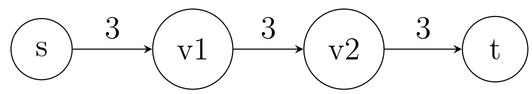
Augmenting path with $f' = 3$



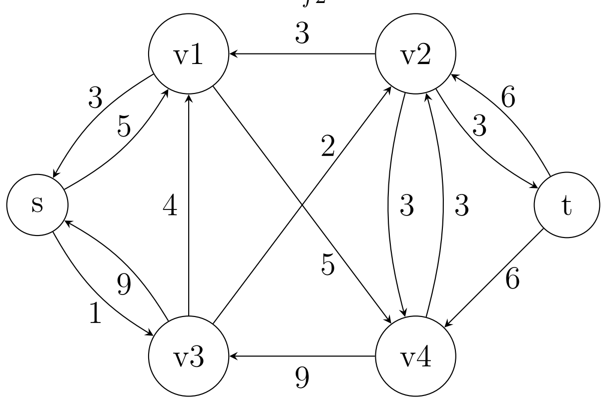
G_{f1}



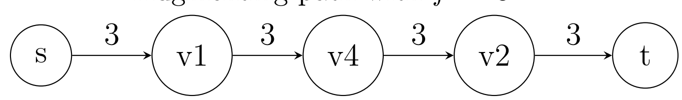
Augmenting path with $f' = 3$



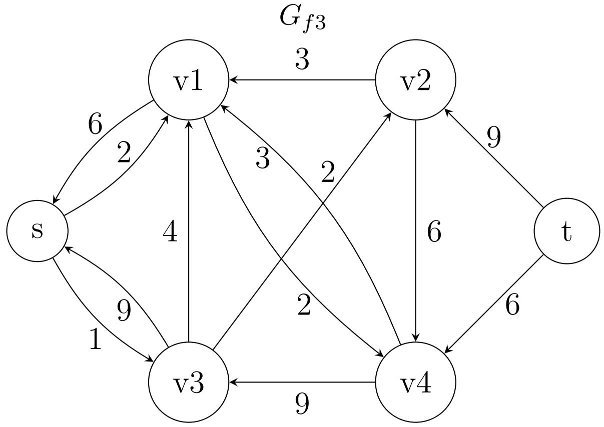
G_{f2}



Augmenting path with $f' = 3$

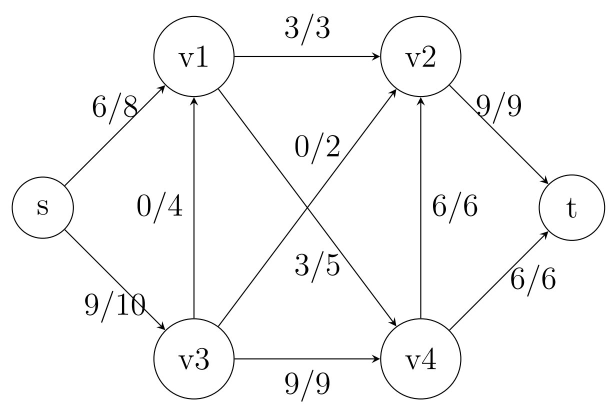


G_{f3}



No more augmenting paths $\rightarrow$ maximum flow

G_3



Problem 3. A linear program is given as follows:

Minimize $x_1 + x_2 + x_3$
subject to:
$2x_1 - 6x_2 + 3x_3 \geq 50$
$3x_1 + 4x_2 - x_3 \geq 40$
$4x_1 - x_2 + 5x_3 \geq 60$
$x_1, x_2 \geq 0$

(a) Express the problem in either standard form or slack form
standard form:
Maximize $-x_1 - x_2 - x_3' + x_3''$
$-2x_1 + 6x_2 - 3x_3' + 3x_3'' \leq -50$
$-3x_1 - 4x_2 + x_3' - x_3'' \leq -40$
$-4x_1 + x_2 - 5x_3' + 5x_3'' \leq -60$
$x_1, x_2, x_3', x_3'' \geq 0$

(b) Solve the problem using a linear programming solver (e.g., linprog in python).

```python
# Import required libraries
import numpy as np
from scipy.optimize import linprog

# Set the inequality constraints matrix
# Note: the inequality constraints must be in the form of <=
A = np.array([[-2, 6, -3, 3], [-3, -4, 1, -1], [-4, 1, -5, 5],
    [-1, 0, 0, 0], [0, -1, 0, 0], [0, 0, -1, 0], [0, 0, 0, -1]])

# Set the inequality constraints vector
b = np.array([-50, -40, -60, 0, 0, 0, 0])

# Set the coefficients of the linear objective function vector
c = np.array([1, 1, 1, -1])

# Solve linear programming problem
res = linprog(c, A_ub=A, b_ub=b)

# Print results
print('Optimal value:', round(res.fun, ndigits=2),
'\nx values:', res.x,
'\nNumber of iterations performed:', res.nit,
'\nStatus:', res.message)

```

Listing 1: homework4.py

```
$python homework4.py
Optimal value: 21.82
x values: [15.45454545  0.          6.36363636  0.        ]
Number of iterations performed: 3
Status: Optimization terminated successfully. (HiGHS Status 7:
    Optimal)
```

Listing 2: execution of homework4.py

(a) What is the optimum solution?
$x_1 + x_2 + x_3 = 21.8181\ldots = \frac{240}{11}$

(b) What are the values of x_1, x_2, and x_3 for this optimum solution?
$x_1 = 15.4545\ldots = \frac{170}{11},\ x_2 = 0,\ x_3 = 6.3636\ldots = \frac{70}{11}$