Problem 3 Code & Result

[1]:
```
import numpy as np
import cvxpy as cvx
import gurobi as grb
```

[2]:
```
h = np.array([10,9,6,6,4,-1])          # player heights (inches above 5'6")
p = cvx.Variable(6, boolean = True)  # 0-1 binary variable for each player i

constraints = [];
constraints += [p[4] + p[5] >= 1]
constraints += [p[1] + p[4] <= 1]
constraints += [p[0] + p[1] <= 1]
constraints += [p[5] + p[1] <= 1]
constraints += [p[5] + p[3] <= 1]
constraints += [sum(p) == 4]

obj = h*p    # Maximizing the height. The maximum average height can be␣
 ↪subsequently be computed.
objective = cvx.Maximize(obj);
```

[3]:
```
prob = cvx.Problem(objective, constraints)
prob.solve(solver = cvx.GUROBI)

print('Problem status: ' + str(prob.status));
if (prob.status == 'optimal'):
    print('Problem value: ' + str(prob.value));
    print('Variable values: ')
    print(p.value)
```

```
Using license file /Users/joehigh/gurobi.lic
Problem status: optimal
Problem value: 26.0
Variable values:
[1. 0. 1. 1. 1. 0.]
```

[4]:
```
max_avg_inch = prob.value/4          # average number of inches above 5ft 6in (4␣
 ↪players)
max_avg_height = prob.value/4 + 66     # 66 inches + max average inches above 5ft␣
 ↪6in
                                       # 5ft 6in = 66 inches

print('Max number of inches above 5ft 6in: ', max_avg_inch, '(inches above 5ft␣
 ↪6in)')
print('Max average height: ', max_avg_height,'inches')
```

```
Max number of inches above 5ft 6in:  6.5 (inches above 5ft 6in)
Max average height:  72.5 inches
```