HW 8

1.) a) $p = 7 \quad q = 17 \quad N = 7 \cdot 17 = 119 \quad \varphi(N) = 6 \cdot 16 = 96$

$e = 5 \quad \gcd(5, 96) = 1$

$5 \cdot d \bmod 96 = 1 \quad d = 77$

pub = (119, 5)
priv = (119, 77)

b) $m = 4^5 \bmod 119$

$m = 72$

$x = 72^{77} \bmod 119$

$x = 4$

c.) $1024^{17} = 1024 \cdot 1024^{16}$

$1024^{16} = (1024^8)^2$

$1024^8 = (1024^4)^2$

$1024^4 = (1024^2)^2$

$1024^2 = (1024^1)^2$

$17 = (10001)_2$

$16 = (10000)_2$

$8 = (1000)_2$

$4 = (100)_2$

$2 = (10)_2$

$\rightarrow 1024 \cdot (((((1024)^2)^2)^2)^2)^2$

$1024 \cdot 1024^{16}$

1024^{17}

Hw 8

Q2.1 Decreasing order of r_i - GCP

Penalty = $r_i \wedge d$

Proof:

OPT: optimal solution that doesn't follow GCP

- so there exists two assignments in OPT where i comes right before j and $r_i < r_j$

- The penalty before i is t and i is completed on day d

  - penalty(i) = $t + r_i^d$
  - penalty(j) = $t + r_i^d + r_j^{d+1}$

if we swap i and j

  - penalty(j)' = $t + r_j^d$
  - penalty(i)' = $t + r_j^d + r_i^{d+1}$

  - penalty(j) $\geq$ penalty(i)'

$$t + r_i^d + r_j^{d+1} \geq t + r_j^d + r_i^{d+1}$$

$$r_j^{d+1} - r_j^d \geq r_i^{d+1} - r_i^d$$

$$d\, r_j^d > d\, r_i^d \quad \text{since } r_i < r_j$$

- Swapping i & j we don't have increase in penalty for optimal solution

- Continue swapping for each inversion in OPT
  - New solution will still be optimal and follow GCP

HW 8
Q3.)

a.) $OPT(i,j)$ = most points on assignments $1 \dots i$ with max of j hours

$$OPT(i,j) = \max\{G_i[j] + OPT(i-1, H-j),\ OPT(i, j-1)\}$$

b.) Base $OPT[0][j] = 0 \quad \forall j \in 0 \dots H$ - init
$OPT[i][0] = G_i[0] \quad \forall i \in 1 \dots n$

```
for i in 1 ... n
    for j in 1 ... H
        OPT[i][j] = max{G_i[j] + OPT[i-1][H-j], OPT[i][j-1]}

return OPT[n][H]
```

c.) Find Path (m) - m is max points returned
P is array on n

```
for i in n ... 1
    for j in 0 ... H
        if m == OPT[i][j]
            P[i] = j
            m = m - G_i[j]
            break inner for loop

return P
```

- P[i] is # of hours to spend on assignment i

HW 8

Q4

a.) OPT(i) = max if selecting i

OPT(i) = i + max{ OPT(i+2), OPT(i+3) }

Go decreasing order

b.) Base OPT[n] = -∞ OPT[n-1] = A[n-1]

OPT[n-2] = A[n-2] - init

for i in n-3 to 0

OPT[i] = i + max{ OPT[i+2], OPT[i+3] }

return max{ OPT[0], OPT[1] }

c.)

find_path( ) -

P is set of indexes

if OPT[0] > OPT[1]

i = 0

else

i = 1

while i <= n

add i to P

if OPT[i+2] > OPT[i+3]

i = i+2

else

i = i+3

return P - is set indexes to follow to get highest value indexes together in order 0/1 ... n