Secondary Structure Prediction

```
# Michael Goldwasser (CSCI 1020)

def w(a,b):
    if {a,b} in ( {'A','U'}, {'C','G'} ):      # complementary bases
        return 1
    else:
        return 0


def buildTable(rna, gap):
    opt = {}
    for g in range(-1,len(rna)):
        # solve for all i,j with j = i + g
        for i in range(len(rna)-g):
            j = i + g
            if g <= gap:
                best = 0    # cannot make any pairs
            else:
                option1 = opt[i+1,j-1] + w(rna[i], rna[j])
                option2 = opt[i+1,j]
                option3 = opt[i,j-1]
                best = max(option1,option2,option3)
                for k in range(i+1,j):
                    if opt[i,k] + opt[k+1,j] > best:
                        best = opt[i,k] + opt[k+1,j]
            opt[i,j] = best
    return opt


def pairs(rna, gap, opt, i, j):
    """Return list of pairs for an optimal pairing."""
    if j - i <= gap:
        return []      # no pairs possible
    elif opt[i,j] == opt[i+1,j]:
        return pairs(rna, gap, opt, i+1, j)
    elif opt[i,j] == opt[i,j-1]:
        return pairs(rna, gap, opt, i, j-1)
    elif opt[i,j] == opt[i+1,j-1] + w(rna[i],rna[j]):
        return [ (i,j) ] + pairs(rna, gap, opt, i+1, j-1)
    else:
        for k in range(i+1,j):
            if opt[i,j] == opt[i,k] + opt[k+1,j]:
                return pairs(rna,gap,opt,i,k) + pairs(rna,gap,opt,k+1,j)
```