18.06 Computational PSet 2 solutions

In high school you may have learned one way to multiply matrices. Chatper 2.4 of your text illustrates many ways.

1. The high school or dot product way appears on pages 67 and pages 68. It says that the entry in the (i, j) position is the sum over k from 1 to n of $a[i, k]b[k, j]$.
2. The matrix times column way appears on the top of page 69.
3. The row times matrix approach, also on the top of page 69.
4. The column times row approach in Example 3 on the top of page 71.

Part 1. The code below implements a basic matrix multiply for square matrices. If you are not using Julia, write one in your language. Either way check a few 3x3 examples, say, and see that it is correct.

In[1]:
```
function matmul_ijk(a,b)
   n=size(a,1)
   c=zeros(a)
   for i=1:n, j=1:n, k=1:n
       c[i,j] +=  a[i,k] * b[k,j]
   end
   c
end
```

Many things that you could do here to check. I ran this code a ton of times:

In[2]:
```
A=rand(0:10,3,3);
B=rand(0:10,3,3);
matmul_ijk(A,B)-A*B
```

Out[2]:
```
3x3 Array{Int64,2}:
 0  0  0
 0  0  0
 0  0  0
```

Part 2: There are six ways to reorder these three loops. Are they all correct matrix multiplies? You could try this or reason about it. Trying it would require all of the permutations:

```
for i=1:n, j=1:n, k=1:n
for i=1:n, k=1:n, j=1:n
for j=1:n, i=1:n, k=1:n
for j=1:n, k=1:n, i=1:n
for k=1:n, i=1:n, j=1:n
for k=1:n, j=1:n, i=1:n
```

Of course there are n^2 dot products. Each one starts or completes a dot product in a different order. In the next part we explore this more carefully.

Part 3: For each of the six ways, decide whether it is of type 1, type 2, type 3, or type 4 repeated here for convenience:

1. The high school or dot product way appears on pages 67 and pages 68. It says that the entry in the (i, j) position is the sum over k from 1 to n of $a[i, k]b[k, j]$.

2. The matrix times column way appears on the top of page 69.

3. The row times matrix approach, also on the top of page 69.

4. The column times row approach in Example 3 on the top of page 71.

Notice that Type 1 means that we are completing every dot product before we start a new one.
Type 2 connotes a column view: that we are partially starting every element in a column, and adding "fairly", until the entire column is complete, at which point we move to the next column.
Type 3 connotes a row view: this is Type 2 with columns replaced by rows.
Type 4 connotes that we start every dot product across the answer, then we add the next product, and so forth.

ijk	Type 1	Dot Products are completed like one reads, left to right, then rows up to down
jik	Type 1	Dot Products are completed first from up to down, and columns left to right
jki	Type 2	In column 1, the first product is computed down the rows, then the second etc. Then we move to the n
ikj	Type 3	In row 1, the first product is computed left to right, then the second. Then we move to the next rows
kij	Type 4	Across the matrix, as one reads, do the first product of the dot product, then add the second, etc.
khi	Type 4	As above, but this time it is down the first column, then the second, etc.