Nathan Tung
CS33 Dis 1C
May 28, 2012

CS33 Homework #5

6.46)

```
void transpose(int *dst, int *src, int dim)
{
        int i, j, x, y, xDim;
        for (i = 0; i < dim; i+=32)
                for (j = 0; j < dim; j+=32)
                        for (x = i; x < (i+32,dim); x++)
                        {
                                xDim = x*dim;
                                for (y = j; y < min(j+32,dim); y++)
                                        dst[x+y*dim] = src[y+xDim];

                                        //dst[j*dim + i] = src[i*dim + j];
                        }
}
```

In this rewritten C-code, I implemented blocking to improve the temporal locality, and thus the efficiency, of the function. By breaking the large array into sub-blocks of size 32 (as in the number of bits in an int), we can treat them like scalars as learned in lecture. This is particularly effective, since blocking is not very dependent on array size. In addition, I added an xDim int so as to prevent the x*dim calculation from looping over and over when the result is always the same.