Assignment Project Exam Help

Cilk and Cilk++
Lecture 2

Add WeChat powcoder

Assignment Project Exam Help

- Fibonacci numbers
- Vector dot product
- Matrix transpose
- Matrix sum
- Matrix product

https://powcoder.com

Add WeChat powcoder

Assignment Project Exam Help

Add WeChat powcoder

fib.c

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <cilk/cilk.h>
#include <cilk/cilk_api.h>

int fib (int n) {
        if (n<2) return n;
        int x = cilk_spawn fib (n-1);
        int y = fib (n-2);
        cilk_sync;
        return x + y;
}
```

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

If these cilk keywords are removed, must yield a correct C program.

Assignment Project Exam Help

fib.c (continued)

Add WeChat powcoder

```
int main (int argc, char* argv[ ] ) {
        int n = argc==1 ? 40 : atoi(argv[1]);
        clock_t start = clock( );
        int result = fib (n);
        clock_t finish = clock( );
        double duration = (double)(finish-start) / CLOCKS_PER_SEC;
        printf ("F(%d) = %d\n", n, result);
        printf ("Calculated in %lf seconds using %d workers\n",
                                duration, __cilkrts_get_nworkers( ));
        return 0;
}
```

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Assignment Project Exam Help

Compile and run

Add WeChat powcoder

```
> cilk  fib.c  –o  fib
> ./fib
> ./fib  40
```

Assignment Project Exam Help

```
F(40) = 102334155
```

https://powcoder.com

```
Calculated in 11.573098 seconds using 24 workers
        (about 0.5 seconds in real time)
```

Add WeChat powcoder

```
> ./fib  45

F(45) = 1134903170
Calculated in 148.570691 seconds using 24 workers
        (about 6 seconds in real time)
```

Computation DAG

Assignment Project Exam Help

```
int fib ( int n ) {
  if (n<2) return n;
  int x = cilk_spawn fib( n - 1 );
  int y = fib( n - 2 );
  cilk_sync;
  return ( x + y );
}
```

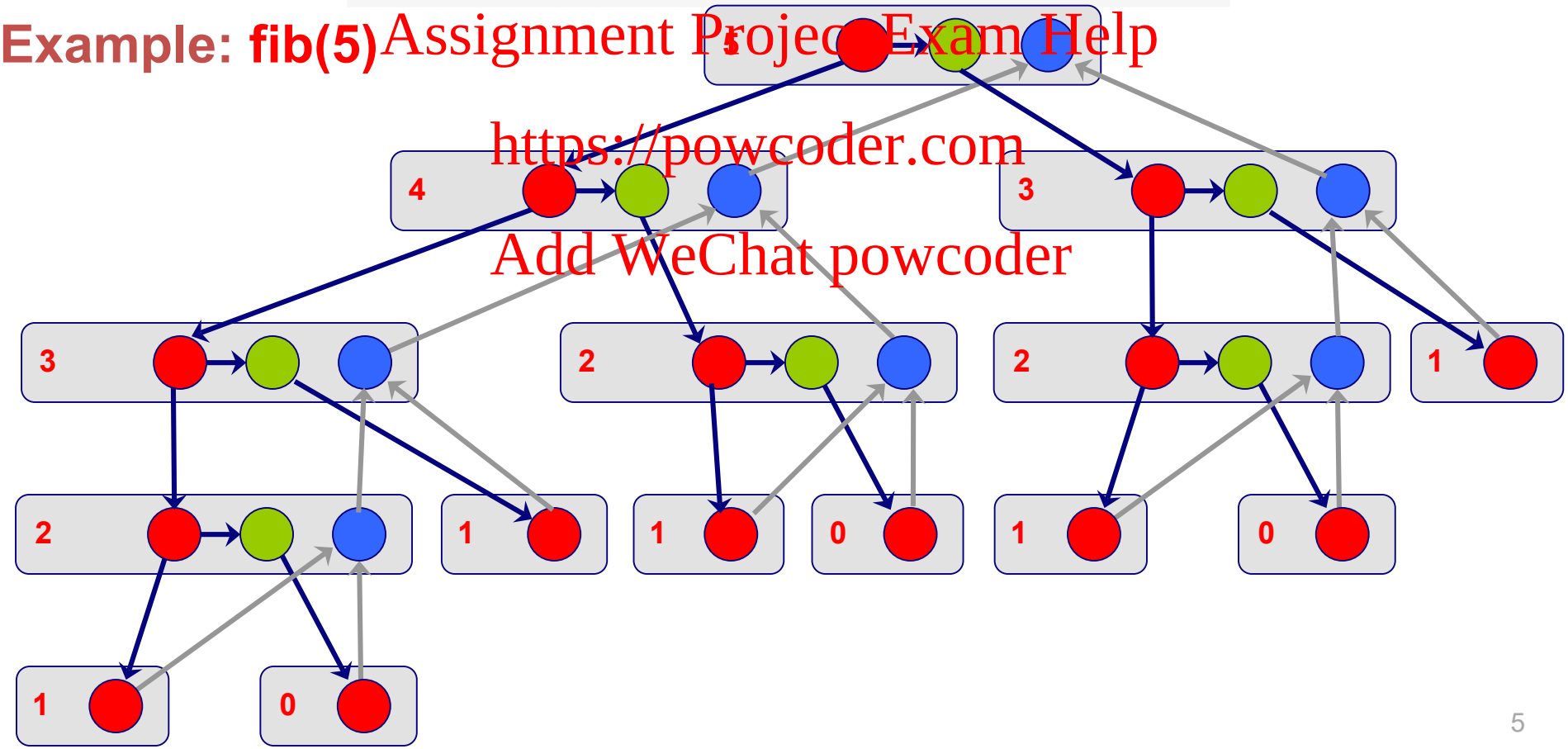
Add WeChat powcoder

Example: fib(5)

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder



Assignment Project Exam Help

dotproduct.c

Add WeChat powcoder

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <cilk/cilk.h>

int dotproduct (int *X, int *Y, int low, int high) {
        if (low==high) return X[low] * Y[low];
        int mid = (low+high)/2;
        int a = cilk_spawn dotproduct (X, Y, low, mid);
        int b = dotproduct (X, Y, mid+1, high);
        cilk_sync;
        return a+b;
}
```

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Assignment Project Exam Help

dotproduct.c (continued)

Add WeChat powcoder

```
int main (int argc, char *argv[ ]) {
        int n = (argc==1)? 100000 : atoi (argv[1]);
        time_t t;
        srand ((unsigned) time(&t));
        int *A = malloc(n * sizeof(int));
        int *B = malloc(n * sizeof(int));
        int k;
        for (k=0; k<n; k++) {                    // initialization (non-parallel)
                A[k] = rand( ) % 100;
                B[k] = rand( ) % 100;
        }
```

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Assignment Project Exam Help

dotproduct.c (continued)

Add WeChat powcoder

Assignment Project Exam Help

```
    clock_t start=clock( );
    int result = dotproduct (A, B, 0, n-1);
    clock_t finish=clock( );
    double duration=(double)(finish-start)/CLOCKS_PER_SEC;
    printf("Result = %d\n", result);
    printf("Dot product of two vectors of size %d took %lf seconds\n",
                                                    n, duration);
    return 0;
}
```

https://powcoder.com

Add WeChat powcoder

Assignment Project Exam Help

Compile and run

Add WeChat powcoder

```
> cilk  dotproduct.c  –o  dotproduct
> ./dotproduct
> ./dotproduct  100000
```

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

```
Result = 244372844
Dot product of two vectors of size 100000 took 0.406831 seconds

> ./dotproduct 4000000

Result = 1215218655
Dot product of two vectors of size 4000000 took 1.391235 seconds
```

Assignment Project Exam Help

transpose.c

Add WeChat powcoder

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <cilk/cilk.h>

void transpose (int **A, int n) {
    int r, c;
    cilk_for (r=1; r<n; r++)
        cilk_for (c=0; c<r; c++) {
            int temp = A[r][c];
            A[r][c] = A[c][r];
            A[c][r] = temp;
        }
}
```

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

If each cilk_for is replaced by for, must yield a correct C program.

Assignment Project Exam Help

transpose.c (continued)

Add WeChat powcoder

```
int main (int argc, char *argv[ ]) {
        int size = (argc==1) ? 100 : atoi (argv[1]), row, col;
        time_t t;
        srand ((unsigned) time(&t));
        int **array = malloc (size * sizeof(int *));
        for (row=0; row<size; row++) {
                array[row] = malloc (size * sizeof(int));
                for (col=0; col<size; col++)
                        array[row][col] = rand( );
        }
```

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Assignment Project Exam Help

transpose.c (continued)

Add WeChat powcoder

Assignment Project Exam Help

```
    clock_t start=clock( );
    transpose (array, size);
    clock_t finish=clock( );
    double duration=(double)(finish-start)/CLOCKS_PER_SEC;
    printf("Transpose of %d-by-%d matrix took %lf seconds\n",
                                size, size, duration);
    return 0;
}
```

https://powcoder.com

Add WeChat powcoder

Assignment Project Exam Help

Compile and run

Add WeChat powcoder

```
> cilk  transpose.c  –o  transpose
> ./transpose
> ./transpose  100
```

Assignment Project Exam Help

https://powcoder.com

```
Transpose of 100-by-100 matrix took 0.067779 seconds
```

Add WeChat powcoder

```
> ./transpose 800

Transpose of 800-by-800 matrix took 0.309467 seconds


> ./transpose 6400

Transpose of 6400-by-6400 matrix took 1.520705 seconds
```

Assignment Project Exam Help

matrixsum.c

Add WeChat powcoder

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <cilk/cilk.h>

void matrixsum (int **X, int **Y, int **Z, int m, int n) {
        int i, j;
        cilk_for (i=0; i<m; i++)
                cilk_for (j=0; j<n; j++)
                        Z[i][j] = X[i][j] + Y[i][j];
}
```

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Assignment Project Exam Help

matrixsum.c (continued)

Add WeChat powcoder

Assignment Project Exam Help

```
int main (int argc, char *argv[ ]) {
        int rows = (argc==1) ? 100 : atoi (argv[1]);
        int cols = (argc<=2) ? 100 : atoi (argv[2]);
        time_t t;
        srand ((unsigned) time(&t));
        int **A = malloc(rows * sizeof(int *));
        int **B = malloc(rows * sizeof(int *));
        int **C = malloc(rows * sizeof(int *));
```

https://powcoder.com

Add WeChat powcoder

Assignment Project Exam Help

matrixsum.c (continued)

Add WeChat powcoder

```
int r, c;
for (r=0; r<rows; r++) {
        A[r] = malloc(cols * sizeof(int));
        B[r] = malloc(cols * sizeof(int));
        C[r] = malloc(cols * sizeof(int));
        for (c=0; c<cols; c++) {
                A[r][c] = rand( ) % 1000;
                B[r][c] = rand( ) % 1000;
        }
}
```

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Assignment Project Exam Help

matrixsum.c (continued)

Add WeChat powcoder

Assignment Project Exam Help

```
    clock_t start=clock( );
    matrixsum (A, B, C, rows, cols);
    clock_t finish=clock( );
    double duration=(double)(finish-start)/CLOCKS_PER_SEC;
    printf("Sum of %d-by-%d matrices took %lf seconds\n",
                                rows, cols, duration);
    return 0;
}
```

https://powcoder.com

Add WeChat powcoder

Assignment Project Exam Help

Compile and run

Add WeChat powcoder

```
> cilk  matrixsum.c  –o  matrixsum
> ./matrixsum
> ./matrixsum  100  160

Sum of 100-by-100 matrices took 0.223749 seconds

> ./matrixsum 600 800

Sum of 600-by-800 matrices took 0.765710 seconds


> ./matrixsum 6400

Sum of 3600-by-6400 matrices took 1.187236 seconds
```

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Parallel Matrix Multiplication

Assignment Project Exam Help

Add WeChat powcoder

```
Seq-MM ( Z, X, Y ) {          // X, Y, Z are n by n matrices

  for (i=0; i<n; i++)
      for (j=0; j<n; j++)
          Z[ i ][ j ] = 0
          for (k=0; k<n; k++)
              Z[ i ][ j ] = Z[ i ][ j ] + X[ i ][ k ] · Y[ k ][ j ]
}
```

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

```
Par-MM ( Z, X, Y ) {          // X, Y, Z are n by n matrices

  cilk_for (i=0; i<n; i++)
      cilk_for (j=0; j<n; j++)
          Z[ i ][ j ] = 0
          for (k=0; k<n; k++)
              Z[ i ][ j ] = Z[ i ][ j ] + X[ i ][ k ] · Y[ k ][ j ]
}
```

Assignment Project Exam Help

Parallel Iterative MM

Add WeChat powcoder

For more speedup, can we also parallelize the innermost loop as follows?

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

```
Par-MM ( Z, X, Y ) {          // X, Y, Z are n by n matrices
  cilk_for (i=0; i<n; i++)
    cilk_for (j=0; j<n; j++)
      Z[ i ][ j ] = 0
      cilk_for (k=0; k<n; k++)
        Z[ i ][ j ] = Z[ i ][ j ] + X[ i ][ k ] · Y[ k ][ j ]
}
```

Assignment Project Exam Help

Parallel Iterative MM

Add WeChat powcoder

```
Par-MM ( Z, X, Y ) {          // X, Y, Z are n by n matrices

  cilk_for (i=0; i<n; i++)
    cilk_for (j=0; j<n; j++)
      Z[ i ][ j ] = 0
      cilk_for (k=0; k<n; k++)
        Z[ i ][ j ] = Z[ i ][ j ] + X[ i ][ k ] · Y[ k ][ j ]
}
```

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

No, this causes a race condition, because n different threads write to the same location Z[i][j].

However, note that the innermost loop computes a dot product. This leads to a better approach…

Assignment Project Exam Help

matrixprod.c

Add WeChat powcoder

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <cilk/cilk.h>

// helper function to replace the innermost loop
int dotproduct (int **X, int **Y, int i, int j, int low, int high) {
        if (low==high) return X[i][low] * Y[low][j];
        int mid = (low+high)/2;
        int a = cilk_spawn dotproduct (X, Y, i, j, low, mid);
        int b = dotproduct (X, Y, i, j, mid+1, high);
        cilk_sync;
        return a+b;
}
```

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Assignment Project Exam Help

matrixprod.c (continued)

Add WeChat powcoder

```
void matrixprod (int **X, int **Y, int **Z, int m, int n, int p) {
        int i, j;
        cilk_for (i=0; i<m; i++)
                cilk_for (j=0; j<p; j++)
                        Z[i][j] = dotproduct (X, Y, i, j, 0, n–1);
}

int main (int argc, char *argv[ ]) {
        int dim1 = (argc==1) ? 1000 : atoi (argv[1]);
        int dim2 = (argc<=2) ? 1000 : atoi (argv[2]);
        int dim3 = (argc<=3) ? 1000 : atoi (argv[3]);
```

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Assignment Project Exam Help

matrixprod.c (continued)

Add WeChat powcoder

```
time_t t;
srand ((unsigned) time(&t));
int **A = malloc(dim1 * sizeof(int *));
int **B = malloc(dim2 * sizeof(int *));
int **C = malloc(dim1 * sizeof(int *));
int r, c;
for (r=0; r<dim1; r++) {
        A[r] = malloc(dim2 * sizeof(int));
        C[r] = malloc(dim3 * sizeof(int));
        for (c=0; c<dim2; c++)
                A[r][c] = rand( ) % 100;
}
```

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Assignment Project Exam Help

matrixprod.c (continued)

Add WeChat powcoder

```
    for (r=0; r<dim2; r++) {
            B[r] = malloc(dim3 * sizeof(int));
            for (c=0; c<dim3; c++)
                    B[r][c] = rand( ) % 100;
    }
    clock_t start=clock( );
    matrixprod (A, B, C, dim1, dim2, dim3);
    clock_t finish=clock( );
    double duration=(double)(finish-start)/CLOCKS_PER_SEC;
    printf("Product of %d-by-%d matrix and %d-by-%d matrix
                    took %lf seconds\n",
            dim1, dim2, dim2, dim3, duration);
    return 0;
}
```

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Assignment Project Exam Help

Compile and run

Add WeChat powcoder

```
> cilk  matrixprod.c  –o  matrixprod
> ./matrixprod
> ./matrixprod  1000  1000  1000
```

Assignment Project Exam Help

Product of 1000-by-1000 matrix and 1000-by-1000 matrix took 99.591898 seconds

https://powcoder.com

(about 4 seconds in real time)

Add WeChat powcoder

```
> ./matrixprod  2000  4000  3000
```

Product of 2000-by-4000 matrix and 4000-by-3000 matrix took 3043.627866 seconds

(about 51 minutes of computation time)

(about 2 minutes in real time)