ompile → gcc -fopenmp BFS.cpp -o BFS
run → ./BFS

DPU

OpenMp :- Open MultiProcessing is an API that supports parallel programming in shared memory system. It simplifies the process of writing parallel code by introducing pragmas which are compiler directives that specify how to parallelize specific sections of code

[#pragma omp parallel for] → this is directive that tells compiler to distribute the iterations of loop across multiple thread allowing for parallel execution.

fopenmp :- This flag tells the compiler to enable Openmp support

Downcommands :-
#pragma omp parallel :- specifies parallel region where multiple thread are created to execute enclosed blocks in parallel.

#pragma omp for :- Distributes iterations of loop among available thread for parallel execution.

#pragma omp critical = defines a critical section of code that can be executed by only one thread at a time

#pragma omp section :- Divides the code block into section that can be executed in parallel by a different threads ~~single only~~.

Ass :-

Bfs :- Bfs stands for breadth first search. It is a graph traversal that explores all vertices of graph in breadth first order i.e visiting all vertices at same level before moving to next level. Bfs start at given vertex (source vertex) and systematically explore its adjacent vertices before moving to next level of vertices.

Bfs :-

1) Start with source vertex & enque it into a queue
2) Mark source vertex as visited.
3) While queue is not empty do following
   1) dequeue the vertex from the front of queue
   2) Process the dequeued vertex
   3) Enqueue all the unvisited neighbours of dequed vertex into queue and mark them as visited
4) Repeat 3rd step until queue is empty.

How Bfs algo using Open mp :-

1) Start with source vertex & enqueue it into shared queue & mark it as visited array or set.
2) Spawn parallel region using 'pragma omp parallel' directive
3) With parallel region create private variable for each thread with private queue
4) Each thread enters loop, & dequeus the vertex from shared queue using critical section.
5) If dequeued vertex is not visited mark it as visited in shared visited array or set.
6) Each thread explores unvisited neighbour of its dequed vertex & enqueue in private queue

7) After all thread have finish exploring combine private queue into shared queue using #pragma omp barrier. Repeat ~~3 to~~ 4 to 7 until all vertices are visited.

8) End Parallel region

Ad	Dis
1) Shortest path finding	1) Memory usage
2) Completness	2) High time complexity
3) visiting all nodes.	3) limited use for weighted graph.

	Time	space
Parallel	O(V+E).	O(V)
serial	O(V+E)	O(V)

Appl^{n} :- Social ~~med~~ nlw analysis, web ~~crawl~~ spiders, Puzzle solving, Graph travel.

DFS :-

DFS :- DFS is a graph traversal algorithm that ~~traver~~ explores the vertices in depth first manner. ~~so bo~~ The algo. start at root node as explores as far as possible along each branch before backtracking. So basic idea is it start from root node and mark node ~~as~~ and move to adjacent unmarked node and continue until no unmarked adjacent node is available. Then backtrack and check for other unmarked ~~no~~ nodes & traverse them.

Algo:-

1. Start with source vertex & mark it as visited.
2. Explore an unvisited neighbour of current vertex.
3. If an unvisited neighbour is found, recursively apply steps 2 and 3, making the neighbour as new ~~the~~ current vertex.
4. If no unvisited node found, backtrack to previous vertex
5. Repeat 2 to 4 until all vertices are visited.

Implement DFS using Openmp.

1. Start with source vertex and mark it as visited.
2. Spawn parallel region using #pragma omp parallel.
3. With parallel Region, create private variable for each thread with private stack
4. Each thread enters loop and pop a vertex from stack. If it is not marked as visited then mark it as visited
5. Each thread explores the unvisited neighbours of pop vertex and pushes them onto stack
6. After all thread have finished exploring their respective vertex, combine stack into shared stack.
7. Repeat from 4 to 6 until all vertices are processed.
8. Exit the parallel region.

Ad	Dis
Simplicity.	If Not complete if stack overflow.
Memory efficiency	
Fetching cycle.	

Appln of Dfs :-
1) Graph Traversal. 3) Decision making
2) Sowing puzzle 4) Backtracking.

Advantage of using parallel programming in Dfs
1) Fast Exploration 2) Utilization of multicore syst(m)
3) Scalibility 4) Efficient resource utilization

complexity O(V+E). O(V).

Ass 2 :-
Bubble sort is simple sorting algo. that repeadetly steps through list compares adjacent elements & swaps them if in wrong order.
Steps :-
1) start with unsorted ~~ass~~ list of element
2) Compare first element & 2nd If 1st is greate than 2nd then swap them.
3) Then compare 2nd & 3rd repeat comparison & sorting.
4) continue this process until last two element of list are compared and swapped.
5) Repeat step 2 to 4 ~~#~~ for remaining passes until no more snap needed.

Parallel bubble sort :-
* Parallelizing the bubble sort involves dividing the sorting task among multiple thread to sort different ~~prop~~ portion of array concurrently.
Each thread perform its own local bubble sort on a subset of array and combine Sorted ~~so~~ subset to obtain final sorted array.

Parallel bubble sort using openmp

1) Start with unsorted array & create a parallel region #pragma omp parallel' directive
2) within the parallel region, initialize flag 'sorted' to track any swap.
3) continue iterating over loop until no swaps are needed.
4) Use loop with #pragma omp for" where each iteration is assigned to each thread. Each thread will perform local Bubble sort.
5) Compare adjacent element & swap them if necessary. Swaping process is performed within a critical section.
6) After each pass set 'sort' to 'true'
7) Once sorting is complete, exit parallel region.

Ad of parallel bubble sort(-)-

1) Faster sorting 2) improved performance
3) Scalability 4) flexibility.

~~Real~~ Example:-

1) Educational Purpose 2) Online Algorithm.
3) Algorithmic Analysis

complexity

Parallel → time $O(\log^2 n)$, space $O(1)$

Merge sort :-

It is popular sorting algo that depends on divide & conquer. This algo work by dividing array into smaller sub array, sorting each sub array & then merging sorted subarray back together to form single sorted array.

Algo :-

1. Divide unsorted array into 2 halves by finding middle index of array
2. Recursively sort two halves by apping merge sort on left and right halves of array separately until each subarray contains one element.
3. Merge sorted subarray :- Merge 2 subarray obtained from previous step to create single sorted array. This done by comparing 2 elements from two sub arrays and placing them in correct order.
4. Copy merged array back to original array.

Parallel Merge Sort :- Parallel merge sort is an extension of merge sort that leverages parallelism to improve performance on multi core or processor by dividing sorting task among multiple thread or processes. (worst).

Parallel Merge sort using Openmp.

1) Divide array among thread use #pragma omp parallel directive to create a parallel region
2) Using #pragma omp for directive to divide the array among thread.
3) Each thread independently sort its assigned subarray.

4) Use #pragma omp barrier directive to ensure that all thread have completed their sorting task
5) Merge sorted array from different thread into single array.

~~Appln of parallel bubble sort~~ merge sort :-
~~1) Educational purpose 2) Small sized array~~
~~3) Integration with other algo.~~

Time = O(nlogn)
Space = O(n)

Appln :-
Data or file merging
External merging

Ad :- 1) Improved Performance
2) Scalibility 3) load balancing.

Min Max Sum Average :-
Parallel Reduction is common technique used in parallel computing to perform a collective computation that combines values from multiple thread or processes into a single result. The goal is to effectively compute average values such as sum, product, maximum, minimum or average across large dataset.

Steps :-
1) Data is partition among the thread or process

2) Each thread or process independently perform local computation on assigned data, typically using sequential reduction algo.
3) ~~Parallel~~ partial results are combined to form new set of partial results. This process continue until there is only one partial result remaining.
4) The final result is obtained by applying same reduction operation to last set of partial results.

Adv:-
1) Improved Performance
2) Flexibility, Scalibility
3) Load balancing

Algo:-
Using #pragma omp parallel for reduction' directive is used to parallelize loop. It specifies that each thread will have its own copy of result variable.

min:-
divide dataset among thread. Each thread finds minimum element in assigned portion of data. Perform parallel reduction by comparing minimum values from each thread and update min with smallest value.

main.
Same

Time :- O(log n)
space :- O(1)

DL :-

ASSI :-

Linear Regression :-
Linear Regression is a statistical modelling technique used to determine relationship between a dependent variable and one or more independent variable. It makes prediction for continous, real numeric variable such as price, sale, age etc. It shows linear relationship which means how it finds value of dependent variable is changing according to value of independent variable

Deep Neural nlw :- DNN is an type of ANN that consists of multiple layers of interconnected nodes called as neurons. They are fundamental component of deep learning, a field of ML and are known for their ability to learn complex pattern and recognition from large amount of data.

Regression :- Regression is statistical method to model relationship between dependent and independent variable with one or more independent variable. It helps to understand how dependent variable is changing corresponding to an independent variable

Appln of Liner regression
1) Economics & finance
2) Predictive Analysis
3) Healthcare & medical research
4) Social Suence
5) Environmental Science

Steps:-

1. Import libraries such as numpy, pandas, tensorflow.
2. load the dataset 'load_boston()' function from sklearn.datasets to load dataset.
3. Preprocess the dataset by spilting it into input feature & target variable
4. Slipt the data into training and testing set
5. Define DNN model using Tensorflow. The model should have multiple layers with appropriate activation function
6. Compile model using loss function & optimizer MSE & Adam.
7. Train model using fit(). specify no. of epochs & batch size for training.
8. Evaluate the model. & predict for new data.

Epoch :-

Epoch refers to complete iteration over entire training dataset during training process. It represents one cycle where model recieves entire training dataset make prediction, computes the loss & update its parameters to improve its performance.

Adam optimizer :- Adam optimizer (Adaptive moment estimation) algorithm commonly used in DL for training neural nw it updates the learning rate for each parameter in nw.

MSE = It calculates average squared difference between predicted values & actual values. It penalizes large errors. $\frac{1}{n} * \Sigma (y - \hat{y})^2$

MAE :- It measures the average absolute difference between predicted value & actual value.

$$MAE = \frac{1}{n} * \Sigma |y - \hat{y}|$$

Appln of DNN

1) NLP, Speech recognization, Recommend^n System, Healthcare, Computer vision.

ASS 1

1) Classification :- It is supervised ML algo. used to classify or identify the category of new observation on basis of training data. It assigns label to the instance of dataset on basis of similarity attributes.

2) Binary classification is fundamental task in ML where goal is to identify or classify data instance into one of the two class or categories. In this, the target variable or label can take any two possible values often referred as positive or negative, 1 or 0, true or false etc. Types logistic regression, SVM, decision tree, Random forest.

Appl^n :- Spam Detection, fraud detection, Disease diagnosis, image classif^n, face recogn^n zation.

ASS 2 :-

CNN :- CNN is deep learning algo commonly used for analyzing visual data such as image & video. Made of multiple layers including convolutional layer, pooling layer & fully connected layer.

convolutional layer, filters are applied to input image. Then this output is passed through pooling layer which downsample the feature maps. This output of pooling layer is then passed to one or more fully connected layer used to make prediction or classify image.

Appln
image classification, object detection, Image Generation, face Recognition, video Analysis.

Panda :- Pandas is an open source library in python for data manipulation and analysis. It provides data structure and functions designed to make work with structured data easy.

Tensorflow :- It is an open source library developed ~~by~~ for numerical computation & ML. It is most popular library ~~developed~~ in field of DL & widely used for building & training various neural nw

Scikit learn :-
It is popular open source library for python It provides wide range of tools & technologies to implement ML & statistic modeling. we can implement regression classification clustering & statistical tool.

matplotlib :- matplotlib is popular data visualization library in python that provides wide range of tools for creating various types of plot, chart & visualization.

seaborn : It is a data visualization library built on top of matplotlib that provides higher level interface for creating attractive & informative statistical graphics

Keras :- Keras is an open source deep learning written in Python. It provides high level API for building and training neural n/w making it easy to prototype and deploy deep learning model.

Dataset :-

1) Boston House :-
Well known dataset & often used to demonstrate regression analysis technique. It consists of 506 houses in Boston, USA. It includs 13 features.

2) IMDB :- It is large collection of movie reviews from IMDB website which is popular source of user generated movie rating & reviews. It consists of 50,000 movie reviews Splits 25000 reviews for training & 25000 reviews for testing. The label for each review is binary 0 indicate negative 1 indicate positive

MNIST Dataset :-
It is collection of 70,000 grayscale images of 28x28 representing 10 different categories of clothing & accessories. The categories includes T-shirt/tops, trousers, pullover, dresses, coats, sandals, shirts, sneakers, bags & ankel boots. Each category has 7000 images.