```
function path = A_star_search(map,MAX_X,MAX_Y)
```

```
%This part is about map/obstacle/and other settings
    %pre-process the grid map, add offset
    size_map = size(map,1);
    Y_offset = 0;
    X_offset = 0;

    %Define the 2D grid map array.
    %Obstacle=-1, Target = 0, Start=1
    MAP=2*(ones(MAX_X,MAX_Y));

    %Initialize MAP with location of the target 最后一个是target
    xval=floor(map(size_map, 1)) + X_offset;
    yval=floor(map(size_map, 2)) + Y_offset;
    xTarget=xval;
    yTarget=yval;
    MAP(xval,yval)=0;

    %Initialize MAP with location of the obstacle 中间的是障碍物
    for i = 2: size_map-1
        xval=floor(map(i, 1)) + X_offset;
        yval=floor(map(i, 2)) + Y_offset;
        MAP(xval,yval)=-1;
    end

    %Initialize MAP with location of the start point 最后一个是起点
    xval=floor(map(1, 1)) + X_offset;
    yval=floor(map(1, 2)) + Y_offset;
    xStart=xval;
    yStart=yval;
    MAP(xval,yval)=1;

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %LISTS USED FOR ALGORITHM
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %OPEN LIST STRUCTURE
    %--------------------------------------------------------------------------
    %IS ON LIST 1/0 |X val |Y val |Parent X val |Parent Y val |h(n) |g(n)|f(n)|
    %--------------------------------------------------------------------------
    OPEN=[];
    %CLOSED LIST STRUCTURE
    %--------------
    %X val | Y val |
    %--------------
    % CLOSED=zeros(MAX_VAL,2);
    CLOSED=[];

    %Put all obstacles on the Closed list
    k=1;%Dummy counter
    for i=1:MAX_X
        for j=1:MAX_Y
            if(MAP(i,j) == -1)
                CLOSED(k,1)=i;
                CLOSED(k,2)=j;
                k=k+1;
            end
        end
    end
    CLOSED_COUNT=size(CLOSED,1);
    %set the starting node as the first node
    xNode=xval;
    yNode=yval;
    OPEN_COUNT=1;
    goal_distance=distance(xNode,yNode,xTarget,yTarget);
    path_cost=0;
    OPEN(OPEN_COUNT,:)=insert_open(xNode,yNode,xNode,yNode,goal_distance,path_cost,goal_distance);
    OPEN(OPEN_COUNT,1)=1;
    CLOSED_COUNT=CLOSED_COUNT+1;
    CLOSED(CLOSED_COUNT,1)=xNode;
    CLOSED(CLOSED_COUNT,2)=yNode;
    NoPath=1;

    MAP

    STORRING_OPEN = [];
    STORRING_OPEN_CNT = 1;
```

输入参数的数目不足。

```
出错 A_star_search (第 5 行)
    size_map = size(map,1);
```

```matlab
%This part is your homework
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% START ALGORITHM
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    path = [xStart,yStart];
    Path_Cnt = 1;

    while(~isempty(OPEN)) %you have to dicide the Conditions for while loop exit finish the while loop
        row = size(OPEN,1);
        i_min = min_fn(OPEN,row,xTarget,yTarget);  % Check min i line in OPEN List
        if i_min == -1
            fprintf("Path Not Available")
            return
        end

        tmp_arrary_min_f = OPEN(i_min,:); % The array can find the min f list
        STORRING_OPEN(STORRING_OPEN_CNT,:) = tmp_arrary_min_f;
        STORRING_OPEN_CNT = STORRING_OPEN_CNT + 1;
        OPEN(i_min,:) = [];
        path(Path_Cnt,:) = [tmp_arrary_min_f(2),tmp_arrary_min_f(3)];     %% Storing the Path
        Path_Cnt = Path_Cnt + 1;



        CLOSED_COUNT=CLOSED_COUNT+1;
        CLOSED(CLOSED_COUNT,1) = tmp_arrary_min_f(2);                      %% put the point already go into the CLOSED LIST
        CLOSED(CLOSED_COUNT,2) = tmp_arrary_min_f(3);                      %% put the point already go into the CLOSED LIST
        CLOSED_COUNT=CLOSED_COUNT+1;
        CLOSED(CLOSED_COUNT,1) = tmp_arrary_min_f(4);
        CLOSED(CLOSED_COUNT,2) = tmp_arrary_min_f(5);

        tmp_exp_array = expand_array(tmp_arrary_min_f(2),tmp_arrary_min_f(3),tmp_arrary_min_f(7),xTarget,yTarget,CLOSED,MAX_X,MAX_Y); % Expand with mi

        % Insert      ||from Node x|| from Node y || h(n) || g(n) || f(n)
        % MIN_F Array ||to Node x  || to Node y |  | h(n) || g(n) || f(n)
        for idex_exp_arr = 1:1:size(tmp_exp_array,1)
            tmp_arr = tmp_exp_array(idex_exp_arr,:);
            tmp_path = [tmp_arr(1),tmp_arr(2)];
            tmp_open = insert_open(tmp_arr(1), ...  % New X Node   %% to Node x,y
                tmp_arr(2), ...                     % New Y Node
                tmp_arrary_min_f(2), ...            % Parent X Node
                tmp_arrary_min_f(3), ...            % Parent Y Node
                tmp_arr(3), ...  % Estimate H
                tmp_arr(4), ...  % Actual Cost G
                tmp_arr(5));     % Heutic f value

            % Check if the OPEN NODE is already in OPEN List
            tmp_open_node = [tmp_open(2),tmp_open(3)];

            tmp_all_open_node = [OPEN(:,2), OPEN(:,3)];

            if (~any(ismember(tmp_all_open_node,tmp_open_node,'rows')))
                OPEN_COUNT = OPEN_COUNT + 1;
                OPEN(OPEN_COUNT,:) = tmp_open;

            elseif (tmp_arr(4) <= tmp_open(7))
                location = ismember(tmp_all_open_node,tmp_open_node,"rows");
                [rowIndex,~] = find(location);
                OPEN(rowIndex,7) = tmp_arr(4);
                OPEN(rowIndex,8) = OPEN(rowIndex,7) + OPEN(rowIndex,6);
                continue;
            end



            if OPEN(OPEN_COUNT,2) == xTarget && OPEN(OPEN_COUNT,3) == yTarget
                path(Path_Cnt,:) = [xTarget,yTarget];
                STORRING_OPEN
                % STORRING_OPEN
                fprintf("SUCCESS FOUND\n")
                return
            end
        end
    end %End of While Loop


        %Once algorithm has run The optimal path is generated by starting of at the
        %last node(if it is the target node) and then identifying its parent node
        %until it reaches the start node.This is the optimal path

        %
        %How to get the optimal path after A_star search?
        %please finish it
        %
```

```
end
```

Published with MATLAB® R2023a