In [50]:
```
## 调用函数库
import numpy as np
import pandas as pd
import geopandas as gpd
import matplotlib.pyplot as plt
import pylab as mpl  #导入中文字体，避免显示乱码
mpl.rcParams['font.sans-serif']=['SimHei']  #设置为黑体字
mpl.rcParams['axes.unicode_minus'] =False
```

In [3]:
```
## 读取数据
df = gpd.read_file("outputs/selected_heights.geojson")
```

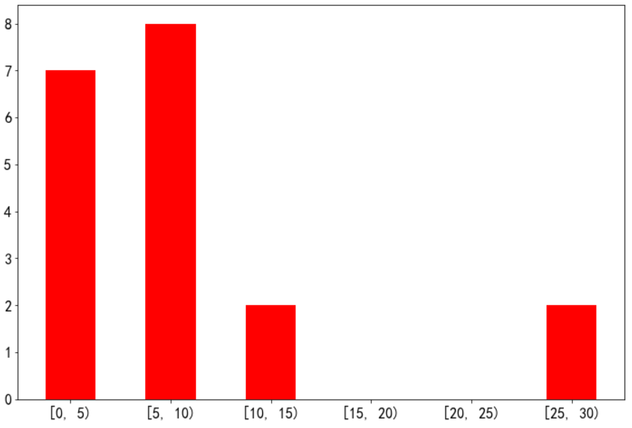
楼层频数统计

In [25]:
```
def frequency_bins(df,bins):
    import pandas as pd
    '''function-频数分布计算'''

    #A-组织数据
    column_name=df.columns[0]
    column_bins_name=df.columns[0]+'_bins'
    df[column_bins_name]=pd.cut(x=df[column_name],bins=bins,right=False) #参数right=False指定为包含左边值，不包括右边值。
    df_bins=df.sort_values(by=[column_name]) #按照分割区间排序
    df_bins.set_index([column_bins_name,df_bins.index],drop=False,inplace=True) #以price_bins和原索引值设置多重索引，同时配置drop=False参数保留原列。
    #print(df_bins.head(10))

    #B-频数计算
    dfBins_frequency=df_bins[column_bins_name].value_counts() #dropna=False
    dfBins_relativeFrequency=df_bins[column_bins_name].value_counts(normalize=True) #参数normalize=True将计算相对频数(次数) dividing all values by the
    dfBins_freqANDrelFreq=pd.DataFrame({'fre':dfBins_frequency,'relFre':dfBins_relativeFrequency})
    #print(dfBins_freqANDrelFreq)

    #C-组中值计算
    df_bins["rating"]=df_bins["rating"].astype(float)
    dfBins_median=df_bins.median(level=0)
    dfBins_median.rename(columns={column_name:'median'},inplace=True)
    #print(dfBins_median)

    #D-合并分割区间、频数计算和组中值的DataFrame格式数据。
    df_fre=dfBins_freqANDrelFreq.join(dfBins_median).sort_index().reset_index() #在合并时会自动匹配index
    #print(ranmen_fre)

    #E-计算频数比例
    df_fre['fre_percent%']=df_fre.apply(lambda row:row['fre']/df_fre.fre.sum()*100,axis=1)

    return df_fre
bins=np.arange(0,31,5) #配置分割区间（组距）
floor_df = pd.DataFrame()
floor_df['rating'] = df['Floor']
floor_fre=frequency_bins(floor_df,bins)
print(floor_fre)
```

```
      index  fre    relFre  median  fre_percent%
0    [0, 5)    7  0.368421     2.0     36.842105
1   [5, 10)    8  0.421053     6.0     42.105263
2  [10, 15)    2  0.105263    12.0     10.526316
3  [15, 20)    0  0.000000     NaN      0.000000
4  [20, 25)    0  0.000000     NaN      0.000000
5  [25, 30)    2  0.105263    27.0     10.526316
```

In [49]:
```
# 画图
plt.figure(figsize=(12,8))
x = range(len(floor_fre))
x_ticks = floor_fre['index']
plt.xticks(x,x_ticks,fontsize = 16)
plt.yticks(fontsize = 16)
plt.bar(x, floor_fre['fre'], width=0.5, color = 'r')
```

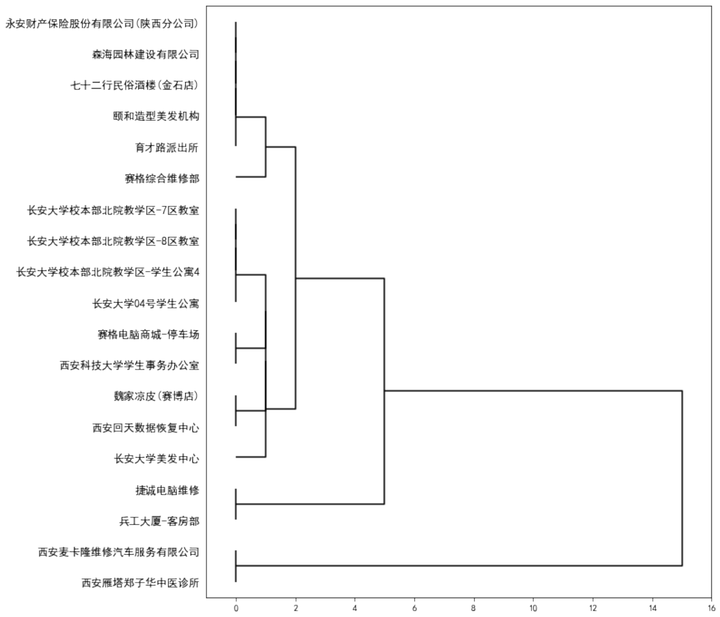
Out[49]:
```
<BarContainer object of 6 artists>
```



由上图可知，所选区域建筑的楼层集中在10层以下，没有15-25层的建筑

楼层层次聚类

In [22]:
```
from scipy.cluster import hierarchy  #用于进行层次聚类，话层次聚类图的工具包

## 重新构造数据
df_cluster = df.loc[:,['Floor','name']]
df_cluster = df_cluster.set_index('name')

plt.figure(figsize=(10,12))
Z = hierarchy.linkage(df_cluster)
hierarchy.dendrogram(Z,labels = df_cluster.index, orientation='right',above_threshold_color='black')
plt.yticks(fontsize= 12)
plt.xlim([-1,16])

plt.show()
```



通过上课可以看出，层次聚类能较好的根据楼层数对建筑进行聚类

In [ ]:
```
```